DCIEM No. CR 1999-11E

~~DCIEM No. 9X-CR-~~

GUIDELINES FOR HUMAN FACTORS ENGINEERING DESIGN AND EVALUATION OF CANADIAN FORCES COMMAND AND CONTROL INFORMATION SYSTEMS

by

Ray Engel*
Engel and Townsend Inc.*
176 Scarborough Road Toronto, Ontario M4E 3M6

Project Manager:
Ray Engel
Telephone no. (416) 698-6376

PWGSC Contract No. W7711-8-7504/001/SVR

on behalf of
DEPARTMENT OF NATIONAL DEFENCE

as represented by
Defence and Civil Institute of Environmental Medicine+
1133 Sheppard Avenue West
North York, Ontario, Canada
M3M 3B9

Acknowledgement

We thank David Beevis for his advice and assistance throughout the project, and for his many helpful suggestions for improving the draft document.

Summary

This document contains guidelines for human factors engineering (HFE) specialists responsible for applying the *Guidelines for Human Factors Engineering Requirements for Canadian Forces Command and Control Information Systems* to the development of a command and control information system (CCIS). This document is also a companion for *User Manual: Guidelines for Human Factors Engineering Requirements for Canadian Forces Command and Control Information Systems* which provides guidelines for project management personnel responsible for managing HFE activities in CCIS development.

The guidelines in this document are in four parts. The first part describes sources of information useful for planning and carrying out HFE work in CCIS development. The second describes how a CCIS acquisition project is designed and executed, and describes the functions that HFE specialists may perform at various stages of a project. The third part, which makes up the document's principal substance, describes HFE analysis, design and testing methods suited to CCIS development, and how application of the different methods meets the requirements in *Guidelines for Human Factors Engineering Requirements for Canadian Forces Command and Control Information Systems*. The final part of the document describes the project management responsibilities of HFE personnel in creating the HFE Plan and managing its execution during the CCIS development project.

CONTENTS

1 INTRODUCTION 1

2 SOURCES OF HFE INFORMATION AND ADVICE 2
2.1 General Reference Books 2
2.1.1 HFE Reference Books 2
2.1.2 Military Command and Control References 4
2.2 HFE Design Standards and Guidelines 4
2.2.1 Human-Computer Interface Design Standards and Guidelines 4
2.2.2 Other HFE Design Standards and Guidelines 6
2.3 HFE Literature Searches 7
2.4 HFE Consulting Assistance 7

3 BACKGROUND OF HFE IN CCIS DEVELOPMENT 8
3.1 HFE in CCIS Development 8
3.1.1 Objectives of HFE in CCIS Development 8
3.1.2 The Nature of Command and Control 9
3.1.3 Difference between HFE in CCIS and HFE in Hardware Systems 11
3.2 The CCIS Project Management Process 12
3.2.1 CCIS Project Design 14
3.2.1.1 Project Organization 15
3.2.1.2 Defining System Requirements from the Statement of Deficiencies 16
3.2.1.3 Investigate Options 17
3.2.1.4 Designing the CCIS Development Project 17
3.2.1.4.1 Acquisition Project Master Plan 18
3.2.1.4.2 Define the System Development Standard and HFE Requirements 19
3.2.1.4.3 Account for the HFE Plan in the Project Budget 21
3.2.1.4.4 Account for the HFE Plan in the Project Schedule 22
3.2.1.4.5 Engage a Contractor 22
3.2.2 Managing the CCIS Development Project 23
3.2.3 Summary of CCIS Project Management 24
3.3 Summary of HFE Background Issues in CCIS Development 25

4 HFE METHODS IN CCIS DEVELOPMENT 27
4.1 Overview of HFE in CCIS Development 28
4.1.1 Formulate Initial Statement of Requirements 28
4.1.2 Account for HFE Work in the Project Budget and Schedule 29
4.1.3 Develop an HFE Plan for Inclusion in Bidding Contractor's Proposal 29
4.1.4 Help the PM Select a Contractor 29
4.1.5 Act as Independent Verification and Validation (IV&V) Agent 29
4.1.6 Conduct HFE Analyses and Tests for Detailed System Requirements 29
4.1.7 Perform HFE Tasks in System Design and Design Testing 30
4.1.8 Participate in Implementation 30
4.1.9 Summary of HFE in Development 31
4.2 HFE Analysis Methods in CCIS Development 31
4.2.1 Note on HFE Analysis Nomenclature 32
4.2.2 Scope of HFE Analyses 33

4.2.3 HFE Mission Analysis 33
4.2.3.1 CCIS HFE Requirements Specifications for Mission Analysis 34
4.2.3.2 Purpose and Nature of Mission Analysis 34
4.2.3.3 The Three Phases of the Command and Control Mission 35
4.2.3.4 Mission Analysis of Planning 36
4.2.3.5 Mission Analysis of Directing 39
4.2.3.6 Mission Analysis of Monitoring 40
4.2.3.6.1 Analysis Based on Weapon System Mission 40
4.2.3.6.2 Analysis by Representative Combat Scenarios 41
4.2.3.7 Mission Analysis of Multilevel Command and Control Systems 44
4.2.3.8 Data Gathering Procedures for Mission Analysis 47
4.2.3.9 HFE Uses of Mission Analysis Information 47
4.2.3.10 Summary of Mission Analysis 48
4.2.4 HFE Function Analysis 49
4.2.4.1 CCIS HFE Requirements Specifications for Function Analysis 49
4.2.4.2 Purpose and Nature of Function Analysis 51
4.2.4.2.1 Purpose and Definition of Function Analysis 51
4.2.4.2.2 Differences Between Function Analysis in HFE and in System Engineering 52
4.2.4.2.3 Function Analysis and Function Allocation 53
4.2.4.2.4 Function Analysis Usage for CCIS Development 53
4.2.4.3 General System Engineering Function Analysis Methods 54
4.2.4.3.1 Function Flow Analysis 55
4.2.4.3.2 Sequence and Timing (SAT) Diagrams 59
4.2.4.3.3 SADT™ and IDEF 60
4.2.4.3.3.1 Structured Analysis and Design Technique (SADT™) 60
4.2.4.3.3.2 IDEF 63
4.2.4.3.4 GRAI Method 63
4.2.4.3.5 Petri Net 66
4.2.4.3.6 Analysis of Functional Organization 66
4.2.4.3.7 Other Function Analysis Methods 71
4.2.4.4 Function Analysis and Phases of the Command and Control Cycle 71
4.2.4.5 Data Gathering Procedures for Function Analysis 72
4.2.4.6 Summary of Function Analysis Methods 72
4.2.5 Task Analysis 73
4.2.5.1 CCIS HFE Requirements Specifications for Task Analysis 76
4.2.5.1.1 General Task Analysis Specifications 76
4.2.5.1.2 Task Analysis Data Specifications 76
4.2.5.2 Structuring Task Analysis Data 78
4.2.5.3 Setting Up a Task Analysis Data Base 79
4.2.5.3.1 Task Analysis Database Logical Structure 80
4.2.5.3.2 Media for Recording and Storing Task Analysis Data 82
4.2.5.4 Procedures for Collecting Task Analysis Data 86
4.2.5.4.1 Task Analysis Procedure for An Existing System 86
4.2.5.4.1.1 Identify Sources of Task Information 87
4.2.5.4.1.2 Acquire and Review Documentation 87
4.2.5.4.1.3 Plan the Data Collection Procedure 87
4.2.5.4.1.4 Construct Data Gathering Instruments 90
4.2.5.4.1.5 Recruit Subject Matter Experts 91

4.2.5.4.1.6 Interview the Subject Matter Experts 92
4.2.5.4.1.7 Summary of Task Analysis Data Collection Procedures 94
4.2.5.4.2 Task Analysis Procedure for A System in Development 95
4.2.5.4.2.1 Setting Up the System Design Task Analysis Database 96
4.2.5.4.2.2 Identify Sources of System Design Task Information 96
4.2.5.4.2.3 Acquire and Review the System Design Documentation 97
4.2.5.4.2.4 Plan the Design Task Analysis Data Collection Procedure 97
4.2.5.4.2.5 Construct Design Task Analysis Data Gathering Instruments 99
4.2.5.4.2.6 Collect the System Design Task Analysis Data 99
4.2.5.4.2.7 Summary of System Design Task Analysis Data Gathering 100
4.2.5.5 Use of Other Task Analysis Methods 100
4.2.5.6 Summary of Task Analysis 101
4.2.6 Analysis by Computer Simulation 103
4.2.6.1 CCIS HFE Requirements Specifications for Computer Simulation Analysis 104
4.2.6.2 System Analysis of Integrated Networks of Tasks (SAINT) 104
4.2.6.3 Service System Models 108
4.2.6.3.1 Critical Path Analysis Methods 108
4.2.6.3.2 Service System Models 109
4.2.6.3.2.1 Analytic Models of Service Systems 109
4.2.6.3.2.2 Purpose-built Simulations of Service Systems 111
4.2.6.3.3 Summary of Operations Management Models 113
4.2.6.4 Simulation by Purpose-built Software 114
4.2.6.4.1 Input Module 115
4.2.6.4.2 Runtime Display 115
4.2.6.4.3 Simulation Program Module 115
4.2.6.4.4 Output Module 116
4.2.6.4.5 Documentation 116
4.2.6.4.6 Summary of Constructing Purpose-built Simulations 117
4.2.6.5 The Role of Operations Statistics in Operational Sequence Analysis 117
4.2.6.5.1 Descriptive Statistics of Operations 117
4.2.6.5.2 Reliability of Simulation Data 118
4.2.6.5.3 Validity of Simulation Data 119
4.2.6.6 Summary of Analysis by Computer Simulation 121
4.2.7 Measures of Performance and Effectiveness 121
4.2.8 Summary of HFE Analysis Methods 123
4.3 HFE Design Methods 124
4.3.1 HFE Design Requirements Methods 124
4.3.1.1 Identification of User Support Requirements 125
4.3.1.2 Identification of Decision Support Requirements 125
4.3.1.2.1 CCIS HFE Requirements Specifications for Decision Support 126
4.3.1.2.2 The Nature of Decision Support in Command and Control 126
4.3.1.2.2.1 Planning Decision Support Applications 127
4.3.1.2.2.2 Decision Support Applications for Monitoring Activities 128
4.3.1.2.3 HFE Role in Identifying Decision Support Requirements 129
4.3.1.3 Function Allocation 130
4.3.1.3.1 CCIS HFE Requirements for Function Allocation 131
4.3.1.3.2 Function Allocation Methods 131
4.3.1.3.2.1 Allocation by Human-Machine Capabilities 132

4.3.1.3.2.2 Allocation by Formal Rating Procedure 133
4.3.1.3.2.3 Allocation to Maximize Technology Support for User Tasks 134
4.3.1.3.2.4 Other Function Allocation Procedures 137
4.3.1.3.3 Roles of HFE in Function Allocation 138
4.3.1.3.4 Summary of Function Allocation 139
4.3.1.4 Tests to Verify Decision Support and Function Allocation Proposals 140
4.3.1.5 Iterative Development of Design Requirements 141
4.3.2 HFE in Detailed Design Activities 143
4.3.2.1 CCIS HFE Requirements Specifications for Integrating HFE in Design 144
4.3.2.2 Maintain Currency of HFE Analyses 145
4.3.2.3 Organizational Factors in Applying HFE Data to Design 145
4.3.2.4 Measures for Making HFE Data Available to Designers 149
4.3.2.4.1 Convenient Access to HFE Information 149
4.3.2.4.1.1 Access to HFE Analysis Data 149
4.3.2.4.1.2 Access to General HFE Information 150
4.3.2.5 Participation in Design Work 151
4.3.2.6 Identify and Apply HFE Design Standards and Guidelines 152
4.3.2.6.1 CCIS HFE Requirements and HFE Standards and Guidelines 152
4.3.2.6.2 Choosing Standards and Guidelines 153
4.3.2.6.2.1 Standard for Equipment, Work Space and Work Station Design 153
4.3.2.6.2.2 Human-computer Interface Design Guidelines 153
4.3.2.6.3 System-specific Style Guide 157
4.3.2.6.4 The role of HFE Specialists in Applying Standards and Guidelines 158
4.3.2.7 HFE Participation in Subsystem Design 158
4.3.2.7.1 HFE in Human-computer Interface Design 159
4.3.2.7.2 HFE in the Design of Equipment, Work Spaces and Work Stations 164
4.3.2.7.3 HFE in Decision Support Application Design 165
4.3.2.7.3.1 Role of the User's Mental Model 165
4.3.2.7.3.2 Display and Confirmation of Input Data 167
4.3.2.7.3.3 Traceability and Verification of Decision Aid Outputs 168
4.3.2.7.3.4 Training and Job Aids for Decision Support 168
4.3.2.7.3.5 User Qualifications for Using Decision Aids 169
4.3.2.7.3.6 Workload Impacts of Decision Support Aids 170
4.3.2.7.3.7 Decision Support Aid Risk Assessment 170
4.3.2.7.4 HFE in Automated Functions Design 171
4.3.2.7.5 HFE in the Production of User Manuals and Training 172
4.3.2.8 HFE in Design Review 173
4.3.2.8.1 CCIS HFE Requirements Specifications for Design Review 173
4.3.2.8.2 HFE Design Review Procedure 173
4.3.2.8.2.1 Compliance with HFE Standards and Guidelines 176
4.3.2.8.2.2 System Function Allocations 176
4.3.2.8.2.3 User Interface Review 177
4.3.2.8.2.4 User Workload Impacts 177
4.3.2.8.2.5 Assessment of System Effectiveness 178
4.3.2.8.2.6 Skill and Knowledge Requirements 179
4.3.2.8.2.7 Provisions for User Manuals and Training 179
4.3.2.8.2.8 Standard Operating Procedures 181
4.3.2.8.3 Scheduling HFE Design Reviews 181
4.3.2.8.4 Summary of HFE Design Review 182

4.3.2.9 Summary of HFE Activities in Design 183
4.4 HFE Testing Methods 184
4.4.1 CCIS HFE Requirements Specifications for HFE Tests 185
4.4.2 An Example of A Design Test 186
4.4.3 HFE Test Plan 189
4.4.4 Specification of Test Measures 191
4.4.4.1 Identifying Candidate Test Measures 191
4.4.4.1.1 Test Measures Specified in the CCIS HFE Requirements 191
4.4.4.1.2 Identification of Workload Measures 194
4.4.4.1.3 Validate Candidate Measures with Test Results Users 196
4.4.4.2 Selecting Measures According to Adequacy of Measurement Properties . . 197
4.4.4.2.1 Validity of Test Measures 197
4.4.4.2.2 Reliability of Test Measures 199
4.4.4.2.3 Validity and Reliability of Different Types of Test Measures 199
4.4.4.2.4 Validity and Reliability of Objective Test Measures 200
4.4.4.2.5 Validity and Reliability of Subjective Test Measures 201
4.4.4.2.5.1 Validity of Subjective Measures 201
4.4.4.2.5.2 Reliability of Subjective Measures 202
4.4.4.2.5.3 Avoiding the Limitations of Subjective Measures 203
4.4.4.3 Final Selection of Test Measures 203
4.4.5 Constructing Test Measurement Instruments 204
4.4.5.1 Instrument Reliability and Design Test Costs 205
4.4.5.2 Construction of Direct Observation Instruments 205
4.4.5.2.1 Data Collection by Observers 206
4.4.5.2.2 Automated Data Recording 206
4.4.5.2.3 Collecting Subject Matter Experts' Evaluations 207
4.4.5.2.4 Data Collection with Video Or Audio Recordings 208
4.4.5.2.5 Summary of Construction of Direct Observation Instruments 209
4.4.5.3 Construction of Questionnaire Instruments 210
4.4.5.3.1 Drafting the Questionnaire 211
4.4.5.3.1.1 Constructing Multiple-choice Items 213
4.4.5.3.1.2 Constructing Rating Scale Items 214
4.4.5.3.1.3 Constructing Point Estimation Items 216
4.4.5.3.1.4 Constructing Rank Order Items 217
4.4.5.3.1.5 Constructing Items in Other Formats 217
4.4.5.3.1.6 Open-ended Items 218
4.4.5.3.1.7 Minimizing Response Bias Effects 218
4.4.5.3.1.8 Formatting the Questionnaire 219
4.4.5.3.2 Questionnaire Review and Pilot Testing 220
4.4.5.3.2.1 Reviewing and Revising the Draft Questionnaire 221
4.4.5.3.2.2 Design the Questionnaire Administration Procedure 221
4.4.5.3.3 Summary of Questionnaire Construction 223
4.4.5.4 Construction of Expert Panel Decision Instruments 223
4.4.5.5 Design Test Medium 226
4.4.5.5.1 Types of Design Test Media 226
4.4.5.5.2 Test Medium Impact on Measurement Validity and Reliability 229
4.4.5.6 Design Test Experimental Design 230
4.4.5.6.1 Determining Sample Size for A Single-variable Measure 231
4.4.5.6.2 Determining Sample Size for A Single-variable Measure 234

4.4.5.6.3 Optimizing the Overall Design 237
4.4.5.6.4 Reconciling Sampling Demands with Project Resources 237
4.4.5.6.5 Summary of Design Test Experimental Design 238
4.4.5.7 Test Operations Scenarios 239
4.4.5.8 Develop Participant Training 241
4.4.5.9 Pilot Test 242
4.4.5.10 Select Test Participants 243
4.4.5.11 Test Schedule 245
4.4.5.12 Provisions for Conducting the Test 246
4.4.5.13 Results Analysis and Reporting 248
4.4.5.13.1 Interpretation of Statistical Confidence Intervals 248
4.4.5.13.2 Analysis of Rating Scale Data 248
4.4.5.13.3 Analysis of Data From Composite Test Measures 249
4.4.5.13.4 Timely Reporting 249
4.4.5.14 Summary of HFE in Design Testing 249
4.5 HFE in Implementation 250
4.6 Summary of HFE Methods 251

5 DEVELOPING AND MANAGING THE HFE PLAN 252
5.1 Constructing an HFE Plan 253
5.1.1 Identification of HFE Plan Requirements 253
5.1.2 HFE Planning Information Requirements 254
5.1.2.1 System Development Plan Requirements 254
5.1.2.2 HFE Plan and System Design Requirements 255
5.1.2.3 Requirements for Level of Effort Information 256
5.1.2.4 Budget and Schedule Constraints 257
5.1.2.5 Summary of HFE Planning Information Requirements 258
5.1.3 Constructing the HFE Plan 259
5.1.3.1 HFE Plan Technical Content 259
5.1.3.2 Integrating the HFE Plan with the System Development Plan 259
5.1.3.3 Integrating HFE with the Project Management Plan 260
5.1.3.4 Constructing the HFE Budget and Schedule 262
5.1.3.4.1 Budgeting for the HFE Plan 262
5.1.3.4.2 HFE Plan Scheduling 263
5.1.3.5 Summary of Constructing the HFE Plan 264
5.2 Managing Execution of the HFE Plan 264
5.2.1 Launching the HFE Plan 265
5.2.2 Controlling the HFE Budget 265
5.2.3 Controlling the HFE Schedule 267
5.2.4 Managing Design Tests and the System Acceptance Trial 267
5.2.5 Manage System Implementation 267
5.2.6 Monitor HFE Reports and Deliverables 267
5.2.7 Conduct Progress Reviews 268
5.2.8 Summary of HFE Project Management Responsibilities 268

6 REFERENCES 269

Tables

Table 1 Summary of system engineering and HFE activities in different types of CCIS development projects. 20
Table 2 Directorate of Business Change Management human factors issues. 35
Table 3 Relationship among system development activities, task analysis activities, and other HFE activities in each phase of system development. 74
Table 4 Comparison of task analysis data base media based on low, medium and high ratings of media on convenience of use features. 86
Table 5 Steps in Meister's procedure for allocating functions. 133
Table 6 Descriptions of equipment and facilities design topics covered in MIL-STD-1472D. 154
Table 7 Human computer design guidelines suitable for application to CCIS development. 155
Table 8 HFE goals of human-computer interface design. 159
Table 9 Requirements for design review checklist item construction. 174
Table 10 Task factors and design features that create workload demand. 177
Table 11 Experimental design for CCIS design test example. 187
Table 12 Steps to select design test measures. 193
Table 13 Example of performance measures with command and control examples. 194
Table 14 Advantages and disadvantages of manual and automated data collection methods. 206
Table 15 Steps for developing a questionnaire. 211
Table 16 Types of questionnaire item formats and examples of their uses. 213
Table 17 Ways to minimize response bias. 218
Table 18 Guidelines for formatting questionnaires. 220
Table 19 Requirements for conducting an interview. 222
Table 20 System representations and their applications in HFE design evaluations. 227
Table 21 Things to do to gain participants' cooperation. 246
Table 22 Summary of HFE tasks in system development. 252

Figures

Figure 1 Steps in the development of a CCIS. 14
Figure 2 Flow chart of steps in a command and control planning process. 37
Figure 3 Critical path of a command and control planning process. 39
Figure 4 Flow chart illustrating headquarters message traffic during monitoring. 44
Figure 5 Optimum control conception of command and control. 46
Figure 6 Example of function flow diagram showing levels of decomposition. 56
Figure 7 Example of function flow analysis using a flow chart representation. 57
Figure 8 Concept of parallel function flow analyses describing different functional dimensions of the same system. 58
Figure 9 Illustration of Sequence and Timing Diagram. 59
Figure 10 Basic SADT™ diagram. 61

Figure 11 Command and control example of items comprising SADT™ inputs, controls, mechanisms and outputs. 61
Figure 12 Illustration of function decomposition in SADT™ 62
Figure 13 Illustration of IDEF 1 data model. 63
Figure 14 Illustration of GRAI method grid (after Figure 12.34 in Luzak [32]). 64
Figure 15 Elements of a Petri net. 67
Figure 16 Illustration of a functional organization chart for a command and control organization. 69
Figure 17 Block diagram illustrating functional relationships. 70
Figure 18 Relationship among system development activities, task analysis activities, and HFE design and testing activities. 75
Figure 19 Example of a telephone dialling task network suitable for a SAINT analysis. . . . 105
Figure 20 Example of Wait Time and Percent Utilization as a function of Processing Time. 110
Figure 21 Histogram of message transmission times from Unit through Battalion to Brigade during a command post exercise. 111
Figure 22 Number of minutes in an hour regulator would be busy as a function of rate of vessels entering sector and times required to service vessels. 112
Figure 23 Flow charts for event driven and time driven simulations. 116
Figure 24 Conception of HFE design activities in a design phase as a design control loop. 182
Figure 25 Example of a rating scale item. 214
Figure 26 Example of a matrix for rating multiple items. 215
Figure 27 Boundaries of the 95% confidence interval for a mean of 50 and a standard deviation of 20. 232
Figure 28 Effect on reliability of increasing or decreasing the length of a composite measurement. 235
Figure 29 Relationship between test exercise headquarters and controllers. 240

1 INTRODUCTION

These guidelines for Human Factors Engineering (HFE) in the design and evaluation of Command, Control and Information Systems (CCIS) were prepared for HFE personnel who are responsible for carrying out the HFE tasks in a CCIS development project.

This document is a companion to two other documents. The first, *Guidelines for Human Factors Engineering Requirements for Canadian Forces Command and Control Information Systems* (CCIS HFE Requirements) [1] advises DND project management staff (PM and PMO) on the requirements for HFE in a CCIS project. It is adapted from *Human Engineering Requirements for Military Systems, Equipment, and Facilities (US MIL-STD-46855)* [2]. The second document, *User Manual: Guidelines for Human Factors Engineering Requirements for Canadian Forces Command and Control Information Systems* (CCIS HFE Requirements User Manual) [3] advises project management staff on how to implement the requirements.

The following table briefly summarizes the contents and uses of the three documents.

CCIS HFE Document	Description
CCIS HFE Requirements	Formally specifies general HFE requirements for a CCIS development project.
CCIS HFE Requirements User Manual	Gives CCIS project management personnel guidance on applying the CCIS HFE Requirements to a specific CCIS project.
CCIS HFE Design Guidelines	Is this document, and gives HFE personnel guidance on creating and conducting an HFE plan that conforms to the CCIS HFE Requirements.

The remainder of this introduction describes the layout of the guidelines. Following this Introduction section, there are four main sections.

Section 2, Sources of HFE Information and Advice, gives directions on where to look for HFE information and HFE consulting help. The section includes the following: 1) an annotated list of general HFE reference books, 2) an annotated list of HFE design standards and guidelines that might apply to a CCIS development project, 3) guidance on how to do a literature search for information on particular HFE topics, and 4) how to find HFE specialist consulting help.

Section 3, HFE Issues in CCIS Development, describes HFE issues that are unique to command and control environments, and unique to the design of a CCIS. The section also describes the acquisition authority project management tasks in a CCIS project.

Section 4, HFE Methods in CCIS Development, describes the HFE methods and activities that go into applying the requirements in the CCIS HFE Requirements to a CCIS project. The section

describes how to perform the three sets of HFE activities that are specified in the CCIS HFE Requirements, and that make up an HFE plan: 1) HFE analyses of users' operations and tasks, 2) HFE activities in system design, and 3) conducting HFE tests of system designs.

Section 5, Developing and Managing the HFE Plan, describes how to comply with the project management requirements in the CCIS HFE Requirements. The section begins with an overview of the project management requirements in the CCIS HFE Requirements, and describes how these requirements apply to the different HFE activities described in Section 4. The section then describes how to use the CCIS HFE Requirements' Data Item Descriptions (DID) to plan and report on the HFE work in a project. The last part of the section describes the acquisition authority's project management needs and expectations.

2 SOURCES OF HFE INFORMATION AND ADVICE

The sources of information in this section are starting points for readers who have little prior experience with CCIS environments. The sources are divided into four categories.

1 *General HFE Reference Books* gives annotated references to a selection of books on HFE topics.

2 *HFE Design Standards and Guidelines* gives references to several sets of human-computer interface design guidelines that would be suitable for specifying the user interface design conventions in a CCIS.

3 *HFE Literature Searches* provides advice on where to look for information on recent HFE research and how to conduct an on-line search for information on HFE subjects.

4 *HFE Consulting Assistance* describes sources of HFE consulting services.

2.1 General Reference Books

This section lists two types of reference books: HFE books, and books on military command and control topics.

2.1.1 HFE Reference Books

The following books cover HFE topics ranging from the elementary to the advanced. If you are an HFE professional, they will likely be already familiar. They are listed here mainly because they are also listed in the CCIS HFE Requirements User Manual - in case you want to be aware of what HFE literature the PM or PMO staff might be reading. As the list suggests by its absence, there is no textbook dedicated to the subject of HFE in CCIS design.

Norman, Donald A. *The Psychology of Everyday Things*. Basic Books, Inc., New York, 257p, 1988 [4]. Though intended as a layman's introduction to the concepts of HFE design, it is also well worth the HFE professional's time. Norman uses everyday objects and devices to illustrate the principles of designing things to make them easy to use, and to show how failure to apply HFE principles produces things that are difficult or impossible to use.

Meister, David. *Conceptual Aspects of Human Factors*. The Johns Hopkins University Press, Baltimore, MD, 263p, 1989 [5]. Meister describes the principles of HFE and the application of HFE to system development, particularly military system development. The book provides good insight into both the practical and theoretical issues associated with applying HFE to system design. Quite apart from its HFE content, the book also provides good insight into the practical realities that HFE practitioners need to be able to deal with in system development environments.

Chapanis, Alphonse. *Human Factors in Systems Engineering*. John Wiley & Sons, Inc. New York. pp. 332, 1996 [6]. The early chapters of this contain a good deal of useful information about system engineering practices, and useful advice on how to make HFE efforts influence system design. The book also contains many examples of HFE applications in system development from Chapanis' own experience.

Salvendy, Gavriel, Ed. *Handbook of Human Factors and Ergonomics*. John Wiley & Sons, 2137p, 1997 [7]. This is the second edition of *Handbook of Human Factors* (listed next). Each chapter in this handbook provides an in-depth technical treatment of a different HFE topic. The topics cover all the major applications of HFE including those relevant to computer systems design. This is probably the best single-reference source for any HFE problem you are likely to encounter in a CCIS development project.

Salvendy, Gavriel (Ed.). *Handbook of Human Factors*. John Wiley & Sons, New York, 1874p, 1987 [8]. This is the first edition of *Handbook of Human Factors and Ergonomics* (listed immediately above). The second addition is more up-to-date, and has broader coverage than this first edition. However, the fact that many of the same HFE topics are covered by different authors in the two editions makes it almost always worthwhile to consult both editions. In fact, for some topics, the older edition may be a better source.

Kantowitz, Barry. H., & Sorkin Robert D. *Human Factors: Understanding People-System Relationships*. John Wiley & Sons New York, 700p, 1983 [9]. This is a well written intermediate level general HFE text. It covers both basic topics such as human perception and learning, and selected HFE applications such as the design of information displays, and work space and equipment design.

Brown, C. Marlin "Lin". *Human-Computer Interface Design*. Ablex Publishing Corp. Norwood, NJ, 236p, 1988 [10]. This book is representative of a number of books that have appeared in recent years to describe the principles and processes underlying human-computer interface design. The discussion provides a general background for applying the human-computer interface design standards and guidelines documents listed in the next section.

2.1.2 Military Command and Control References

The references in the preceding section (Section 2.1) were given in the CCIS HFE Requirements User Manual to provide PMs and PMO staff, who are normally military personnel, with some background in HFE topics. The following reference will provide HFE personnel with some complementary background in military command and control topics.

Fuller, J. F. C. *A Military History of the Western World*, Vols. I to III. Da Capo Press, New York, 1987 (original publication 1954). [11]. This is a history of decisive battles from early times to World War II. Fuller began writing it at Camberley Staff College where he says that the history of war was a subject little studied by the people who ought to have the greatest need to know it. Fuller's history shows the critical role that command and control matters play in battle outcomes, and the unique and enduring nature of command and control methods and organizations.

There is a large body of literature on military history and military topics, and it is easy to find a book or article on almost any subject the reader is interested in.

2.2 HFE Design Standards and Guidelines

This section contains annotated references to HFE Standards and Guidelines that can, or in some cases, ought to be applied to CCIS developments to help ensure that the system has good HFE design. The references are divided into two groups. The first are human-computer interface design guidelines. One or more of these documents can be used to guide the design of the CCIS human-computer interface. The second group of references are standards and guidelines that may be applicable to other aspects of the CCIS design: the design of work stations and work spaces for example.

In recent years, military standards and guidelines have been undergoing reviews and revisions aimed at simplifying their application to acquisition projects. Much of the revision work has been designed to make standards and guidelines less constraining on design innovation, and less demanding of time spent on meeting project management reporting requirements. By the time this is read, the contents the US DoD standards may appear under different names, or be in different forms, or even have disappeared. However, whether or not a set of standards or guidelines has official status does not affect its potential usefulness for guiding design in a particular project.

2.2.1 Human-Computer Interface Design Standards and Guidelines

The following references are a selection of human-computer interface design standards and guidelines that give direction for designing the elements and functions of a human-computer interface. The first reference is a military standard which an interface design should conform with

in any CCIS development. The remaining references are guidelines that provide empirically-based advice for designing specific interface elements.

U.S. Department of Defense. *Human engineering design criteria for military systems, equipment, and facilities (MIL-STD 1472D)*[1]. U.S. Department of Defense, Washington, DC, March, 1989 [12]. MIL-STD-1472D includes standards for computer-user interface design. The standards cover computer displays, keyboards, pointing devices, system response times, and user-computer interaction procedures. Since MIL-STD-1472D covers only basic human-computer interface design requirements, it will not cover all the interface design questions likely to come up in a particular CCIS design. It should therefore be supplemented with one or more of the sets of guidelines listed next.

Open Software Foundation. *OSF/Motif Style Guide, Revision 1.0*. Open Software Foundation, Cambridge, Massachusetts, 1989 [13]. This document describes windows system concepts and gives guidelines for using input and output devices, and designing windows displays, controls, menus and interfaces with client applications. The OSF/Motif guidelines are restricted to ensuring that OSF/Motif applications reflect OSF/Motif conventions. As a result, they do not cover many general interface design topics that are included in other interface design guidelines.

Avery, Larry W. & Bowser, Samuel E. (Eds.). *Human Factors Design Guidelines for the Army Tactical Command and Control System (ATCCS) Soldier-Machine Interface: Version 2.0.* Prepared by Battelle Pacific Northwest Laboratory, U.S. Dept. Of Energy, for U. S. Army Tactical Command and Control System Experimentation Site, Fort Lewis, Washington, May 1992 [14]. This is the most comprehensive of any set of general interface design guidelines. Topics covered include: interactive control, form filling operations, function keys applications, menu usage and navigation, direct manipulation concepts, windows use and design, maps and situation displays, presentation graphics, screen design, colour usage, decision aids, query language, on-line help, embedded training, and special displays.

CTA Incorporated. *User-Interface Guidelines.* Data Systems Technology Division, Goddard Space Flight Center, January 1996 [15]. These guidelines are oriented to windows-based interfaces. They cover principles of user-centred interface design; use of interface components such as cursors, text fields, and command buttons; screen layout including text and table displays; user input dialogues and data security; windows design and manipulation; visual coding using effects such as colour and brightness; and the design of help systems and error messages. There is no guidance on graphics displays such as data charts, bitmap displays, or map displays.

U. S. Department of Defense. *Department of Defense Handbook: Human Engineering Guidelines for Management Information Systems (MIL-HDBK-761A).* U.S. Department of Defense, Washington, DC, 272p, September, 1989 [16]. These guidelines cover dialogue control, data entry, data display, job performance aids, data communication, and data protection. There is also some material on the interface design process including how to do a function and task

[1] MIL-STD-1472 has been revised to MIL-STD-1472E. However, a copy was not available at the time of writing this document.

analyses for the interface design. There is also a small amount of material on expert systems. The examples used in the guidelines are predominantly tactical missile command and control applications.

2.2.2 Other HFE Design Standards and Guidelines

The following documents contain standards and guidelines for components of system design that are beyond the boundaries of human-computer interface design.

U.S. Department of Defense. *Human Engineering Requirements for Military Systems, Equipment, and Facilities (MIL-STD-46855)*[2]. U.S. Department of Defense, Washington, DC, 1994 [2]. The CCIS HFE Requirements is modelled on this standard. The CCIS HFE Requirements is applicable to a predominantly computer-based command and control information system, while MIL-STD-46855 is intended for the development of hardware systems such as weapons systems, vehicles, and the like. If the CCIS to be developed contains hardware components requiring significant development effort, MIL-STD-46855, or applicable parts of it should be included in the CCIS project's HFE requirements.

U.S. Department of Defense. *Human engineering design criteria for military systems, equipment, and facilities (MIL-STD 1472D)*. U.S. Department of Defense, Washington, DC, March 1989 [12]. This standard is also cited in the previous section listing human-computer interface design standards and guidelines. Along with human-computer interface design standards, MIL-STD-1472D also contains HFE design standards that may be applicable to a particular CCIS. These other standards specify design criteria for the design of purpose-built displays, control consoles, warning systems, and other equipment and facilities. If the CCIS includes such components, these MIL-STD-1472D criteria can be used to define their design requirements.

Engel, R. & Townsend M. *Guidelines for the Design and Evaluation of Operator Interfaces for Computer Based Control Systems*. For Defence and Civil Institute of Environmental Medicine under Contract # 09SE.W7711-5-9362/01-SE. Engel and Townsend, Toronto, Ont. 1989 [17]. These guidelines describe how to manage and carry out the HFE development and evaluation of a human-computer interface design for a computer-based control system. Much of the material in Section 4 (HFE Methods in CCIS Development) of this CCIS HFE Design Guidelines is an updated version of material in the HFE Guidelines for Control Systems. Since the HFE Guidelines for Control Systems are for computer-based control of hardware systems, the material will also apply to CCISs that include hardware components.

Beevis, D., Bost, R. Döring, B., Nordø, E., Oberman, F., Papin, J-P, Streets, D., & Schuffel, H. *Analysis Techniques for Man-Machine System Design, Vols 1 & 2*. NATO Defence Research Group Technical Report Ac/243(Panel 8)TR/7, Defence and Civil Institute of Environmental

[2] MIL-STD-46855 has been revised to MIL-HDBK-46855. However, a copy was not available at the time of writing this document.

Medicine, Downsview, Ontario, Canada, 104p, July 1992 [18]. This document contains systematic descriptions and critical reviews of all the significant HFE analysis methods, including those relevant to CCIS development. Although the contents are oriented to human-machine interface design problems, most of the principles of the analyses will apply to CCIS design problems. The document is particularly useful for getting a systematic overview of a set of candidate analysis methods, and then either applying the method, or finding additional information on how to apply the method.

2.3 HFE Literature Searches

The following periodicals are primary sources for recent research on HFE topics that are likely to apply to a CCIS development project.

Human Factors.
Applied Ergonomics.
IEEE Transactions on Systems, Man and Cybernetics.
International Journal of Human-Computer Studies (formerly International Journal of Man-Machine Studies).

HFE literature can also be found by searching on-line data bases.

The Defence and Civil Institute of Environmental Medicine (DCIEM) can also provide assistance in literature searches (see the next section). DCIEM's HFE specialists can assist by advising on a search strategy, conducting a search, and often, by providing references to, or copies of the pertinent documents.

2.4 HFE Consulting Assistance

The field of HFE is considerably broader than any one HFE practitioner's training and experience can encompass. Therefore, no HFE practitioner need have any qualms about looking for other HFE consulting help when confronted by a problem that outside his or her area of expertise. Indeed, to do otherwise in this circumstance is unprofessional.

Any search for HFE consulting advice on CCIS topics should begin with an enquiry to DCIEM at the address shown on the right. The staff of DCIEM provide HFE consulting services to CF branches and DND Directorates. These services include *ad hoc* consultation on particular problems, assisting with developing the HFE components of Statements of Work, assisting in finding or selecting HFE contractors, and conducting HFE

Defence and Civil Institute of Environmental Medicine
1133 Sheppard Avenue. West.
PO Box 2000
North York, Ontario
M3M 3B9

research projects. DCIEM can provide the names of individuals and firms that do HFE work. However, DCIEM cannot guarantee or be in any way responsible for the quality of the work done by the firms whose names it might suggest.

3 BACKGROUND OF HFE IN CCIS DEVELOPMENT

This section has two parts. The first describes the nature of HFE work in CCIS development, particularly how it differs from HFE work in more traditional hardware system development. This material is intended for readers who are new to HFE applications in CCIS development. This material can be safely skipped over by readers who are already familiar with CCIS environments.

The second part of this section describes the acquisition authority project management activities that go into designing a CCIS development project. This material is intended to give readers who are not acquainted with DND system acquisition and project management processes some insight into how acquisition and project management considerations affect HFE activities in CCIS development. Readers who are already with DND acquisition and project management procedures can also safely skip over this material.

3.1 HFE in CCIS Development

This section outlines the general role of HFE in CCIS development. The topics that will be covered are:

1 the objectives of HFE in CCIS development,

2 the nature of command and control, and the role of a CCIS in command and control, and

3 the differences between HFE activities in CCIS development and HFE in other types of system development.

3.1.1 Objectives of HFE in CCIS Development

The central objective of HFE in CCIS development is generally the same as the objective in other types of system development, that is, to make the system safe and easy for people to use. This includes producing a system design that:

1 lets users perform productive tasks quickly and efficiently;

2 makes it easy for users to exploit the full potential of the system;

3 minimizes needs for users to perform repetitive mechanical tasks;

4 minimizes the chances of users making errors in using or operating the system, and

5 gives users satisfaction from using the system.

There are several organizational benefits that flow from accomplishing these objectives:

1 maximization of the safety of operations;

2 maximization of personnel productivity;

3 minimization of personnel overhead costs caused by unnecessary numbers of personnel, or needs for personnel with expensive qualifications;

4 minimization of personnel training costs associated with unnecessarily complex system operating procedures, and

5 minimization of personnel overhead costs associated with low job satisfaction.

Although these two sets of goals are common to all HFE work, their achievement lies in adapting HFE methods to the demands of each particular application. The following sections describe the demands that distinguish HFE work in CCIS development from other types of HFE work. The first of the following sections describes how the general nature of command and control influences HFE work. Subsequent sections will focus on HFE technical issues in CCIS development.

3.1.2 The Nature of Command and Control

Command and control has two components. One is the set of principles that defines command and control as a concept. The other is the command and control system, which is the organization that a commander uses to exercise command and control over military forces.

There are many definitions of command and control, and every military organization has its own formal definition. The definition that will be used here is a common one that is convenient for discussing HFE issues in command and control. This definition is based on one used in [19]. *Command and control is the commander's use of command authority to direct the operations of the forces under the commander's control.* The term command authority means that a commander is totally responsible for the operations of his or her forces. And to be able to exercise this responsibility, a commander also has near absolute authority over these forces.

Every commander exercises command and control by means of a command and control system. A command and control system contains three components:

1 the commanders and staff who plan, direct and monitor operations;

2 the military doctrine and procedures that commanders and staff comply with while performing command and control activities, and

3 the facilities and equipment that support command and control operations.

A CCIS is part of the facilities and equipment that make up the third component.

The first part of this definition of a command and control system also coincidentally defines the three phases of command and control activities in a military operation: 1) planning the operation, 2) directing the operation, and 3) monitoring the execution of the operation. Planning consists of preparing the orders that the forces will follow to execute the operation. Directing is the process

of issuing these orders. Monitoring is the process of tracking the progress of the operation once it gets under way, and if necessary, altering the operation plan to respond to unanticipated problems[3].

Things that go wrong in command and control are almost always associated with either the planning phase or the monitoring phase. A plan can be inadequate because it is tactically wrong, or because it is administratively slipshod. Things that go wrong during the monitoring phase are almost always things that are caused by lack of information about the progress of operations, or long delays in getting information about the progress of operations. The directing phase of command and control is the normally straightforward mechanical process of distributing orders - straightforward provided that the communications network for distributing orders is intact.

The definition of command and control implies two principles to keep in mind whenever one makes some system change aimed at improving the performance of the command and control system. The principles are these.

1 Ultimately, the system change must lead to an improvement in the commander's command and control effectiveness by improving the ability of commanders and staff to plan, direct or monitor operations.

2 The system change must preserve the integrity of the doctrine and procedures that define the principles of command and control and the logical structure of the command and control system.

As these principles imply, the performance of a command and control system is ultimately a function of the performance of the human elements of the command and control system: that is, the commanders and staff. It is because command and control is fundamentally a matter of human performance that HFE has a critical role to play in command and control systems development.

The general definition of a command and control system also defines the functions that a CCIS performs in command and control. The CCIS is made up of the equipment and facilities that commanders and their staffs use to communicate among themselves, and use to access, store and manipulate the tactical information that they use to plan, direct and monitor operations. Modern CCISs are almost universally computer-based communications and information systems. For this reason, the bulk of the CCIS HFE Requirements is oriented to computer system design issues, and issues of how humans use computers. Although computer systems are at the core of a CCIS, the facilities that house these computers and their users are also part of the command and control system. Several of the standards cited in Section 2.2 (page 4), particularly MIL-STD-46855 [2] (or its successor, MIL-HDBK-46855) and MIL-STD-1472D [12], provide HFE standards for such facilities.

The principles that define command and control, and command and control systems, also define the unique characteristics of HFE work in command and control environments. The next section

[3] The planning, directing and monitoring conception is widely used in the CF in command and control work.

describes these characteristics in more detail by comparing HFE activities in command and control system development with HFE work in traditional hardware system development.

3.1.3 Difference between HFE in CCIS and HFE in Hardware Systems

The nature of HFE activities in CCIS development is easiest to see by contrasting CCIS development with the tradition of hardware system development from which modern HFE has emerged. From the design perspective, the essential difference between a CCIS and a military hardware system is that in a CCIS, the system has to support the user, while in a hardware system, the user (operator) supports the system.

In an element of a military force, whether the element is as small as an army platoon or as large as naval fleet, the entire function of the element's command and control organization is to support the commander's control over the element's resources. Similarly, the CCIS being part of the command and control system, its entire function is to support the commander's command and control needs. As outlined in the previous section, the CCIS does this by serving the communications and information processing needs of the commanders and staff who make up the command and control organization. In short, in a CCIS, the first priority of the system is to support users' tasks. In military hardware systems, the human-system relationship is reversed.

In a military hardware system, for example, an armoured vehicle, the humans in the system are operators whose functions are to act as system control elements. As operators, the humans in a hardware system are essentially subsystems that have the same operational status as any of the system's other subsystems. The difference between the roles of humans in CCISs and hardware systems results in two differences between HFE in CCIS development and hardware system development. The first difference is in the stage of system development at which substantial HFE work begins. The second difference is in the nature of the HFE work that is done in the two types of systems.

In a military hardware system, design normally begins with the identifying the tactical role that the system is to fulfil. In an armoured vehicle for example, identifying the tactical role leads to defining specifications for subsystems such as hull configuration, propulsion system, weapons, sensors, communications, and finally, the crew who will operate the vehicle. In this process, much of the system development will be done before much HFE work needs to be done. For example, the first major HFE work in the armoured vehicle's development will be to allocate functions to the crew, which can occur only after the major hardware systems have been identified and have gone through at least preliminary design.

In CCIS development, in contrast to hardware system development, not even preliminary design can start until there is an (HFE) analysis of the functions and tasks that command and control personnel perform. In short, in a CCIS development, HFE work has to begin at the beginning development. In a hardware system development, the main HFE work cannot start until the early system design work has been done.

The different characteristics of HFE work in hardware systems and HFE work in CCISs come largely from differences in the tasks that humans do in the two types of system. In hardware systems, the tasks are predominantly machine operating tasks, so the emphasis of the HFE work will be on the effectiveness of the human-system interface, and on ensuring that the humans in the system make a maximum contribution to the system's ability to accomplish its tactical mission. In contrast, the command and control tasks that CCIS users do cover a considerably wider range of human behaviour and human capabilities than common machine operating tasks do. In command and control settings, users' tasks include a wide variety of complex problem solving and decision making tasks in addition to basic system operating tasks. Also in contrast to hardware system HFE, CCIS design must concentrate not just on ensuring that the system design maximizes users' ability to perform system operating tasks, but also on users' ability to perform a variety of complex tasks that depend on the functions provided by the system.

The CCIS HFE Requirements derivation from the hardware oriented MIL-STD-46855 [2] was specifically designed to reflect the differences between CCIS development and hardware system development, particularly the earlier start of HFE work in CCIS development, and the need to ensure that the CCIS system design supports users' tasks. The CCIS HFE Requirements similarly reflects the need for the HFE plan in CCIS development has to be more deeply integrated into the overall system engineering[4] effort than is typical in hardware system development.

The closer integration between HFE and other system engineering activities in turn means that HFE personnel in a CCIS development have to be more involved in the overall system development project management process than is typical in hardware system development. The next section outlines the general CF CCIS system development process and how applying the CCIS HFE Requirements fits into this process.

3.2 The CCIS Project Management Process

The CCIS HFE Requirements provide the system development contractor's[5] HFE personnel with specifications for planning, managing and doing the HFE component of the CCIS development work. From a contractor's perspective, the application of the CCIS HFE Requirements starts when the contractor begins preparing a proposal in response to the acquisition authority's Request for Proposal (RFP) and Statement of Work (SOW). In fact, however, by the time an RFP is issued, a great deal of work will have been done by the PM and PMO staff to prepare the SOW that the contractor will use as specifications for a CCIS development plan. As this implies, a CCIS development project has two major phases. The first is the project design phase which

---

[4] Although the definition of system engineering in the CCIS HFE Requirements 3.1 includes HFE as part of system engineering, the term system engineering will often be used here in a context that means one or more of the non-HFE disciplines that make up system engineering.

[5] In this document, the term contractor means any organization performing the CCIS development work irrespective of whether the organization is a private company, DND in-house establishment, or other agency.

covers the activities in the acquisition authority PMO leading to appointing a contractor. The second is the project execution phase in which the contractor does the actual system development.

The following lists the steps in each of the two main phases of a CCIS development project.

1 Project design.

- 1.1 The user organization identifies the need for a new system by creating a statement of deficiencies describing the deficiencies in the existing system.
- 1.2 A PMO is created, and the PM translates the user's statement of deficiencies into a statement of system requirements.
- 1.3 The PM identifies and evaluates options for acquiring a system that will satisfy the requirements, and in consultation with the user and others, selects an acquisition option.
- 1.4 The PM develops the acquisition project by specifying the system development approach, project budget and schedule, and issuing an RFP.
- 1.5 A contractor is hired to execute the project.

2 Project execution.

- 2.1 The contractor develops a detailed set of system requirements and specifications.
- 2.2 The contractor develops a system design and tests the design to see if the design will satisfy the system requirements and specifications.
- 2.3 Subject to the results of the design tests, the contractor continues to revise and test the design until the design satisfies the user's requirements.
- 2.4 The contractor produces the system and helps the user to implement it.

Figure 1 shows a flow diagram that summarizes the two project phases and the steps within each phase. Although the execution phase of a project is shown in Figure 1 as a set of contractor activities, the acquisition authority PM and PMO staff remain deeply involved in the project, but now in the role of monitoring and controlling the execution of the project.

Although Figure 1 suggests a continuous system development process that goes from identification of deficiencies through to an implemented system all in one integrated project, in practice, development of a fielded system will come from a series of projects, each producing a more operational version of the final system. Section 3.2.1.4.1 (page 18) later on describes how a system's development is typically broken up into sub-projects, and how HFE activities vary with different stages of development.

To intelligently apply the CCIS HFE Requirements to a project, the HFE specialist needs to have not only an understanding of how to mange the HFE development work itself, but also an understanding of what went into designing the project in the first place. The section immediately following this one will describe the project design phase. A subsequent section will describe the project execution phase.

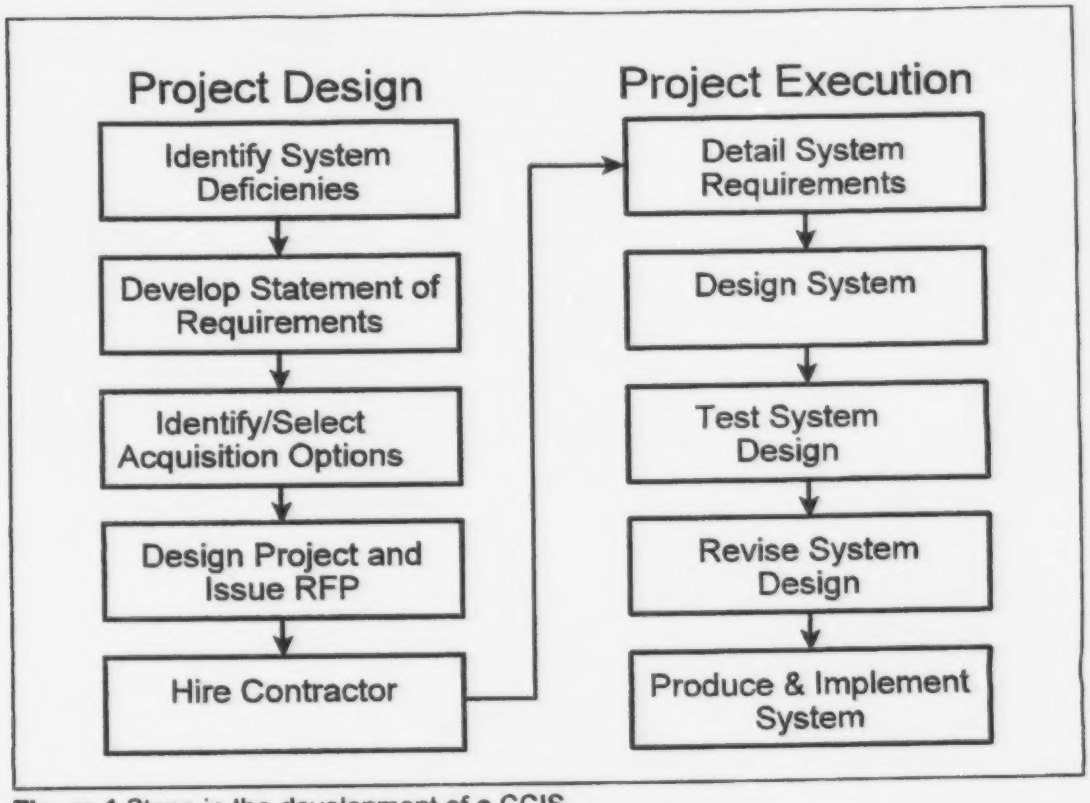

Figure 1 Steps in the development of a CCIS.

3.2.1 CCIS Project Design

The following expands the four project design steps in Figure 1 into the detailed steps that the user organization and the PMO follow to design a project. The steps are the following.

1 The user organization identifies an operational need for a new system, normally by expressing the need in a statement of an existing system's deficiencies.

2 The requisitions directorate sets up a project with a PM and PMO to acquire a new system that will remedy the deficiencies of the existing system.

3 The PM investigates options for remedying the deficiencies. Options typically include three possibilities: 1) buy an off-the-shelf system, 2) hire a contractor to develop a new system, and 3) do a research and development project to explore one or both of the first two options.

4 The user organization and related authorities select the best option.

5 Once the acquisition option has been chosen, the PM designs the project. This includes five steps:

5.1 converting the user's statement of deficiencies into a statement of system requirements;

5.2 preparing a project budget and schedule;

5.3 defining a project management method for conducting the system development project;

5.4 preparing an RFP containing an SOW and issuing it to potential bidders, and

5.5 selecting the system development contractor (in consultation with other authorities).

The following subsections briefly describe what happens in each of these steps, particularly the things that need to be kept in mind when drafting or carrying out an HFE plan.

3.2.1.1 Project Organization

This subsection covers three steps in getting a CCIS development under way: 1) setting up a PMO, 2) investigating acquisition options for dealing with the user organization's requirements, and 3) selecting the best option for meeting the user's requirements.

There are two parties who have primary roles in an acquisition project, the user organization and a DND requirements directorate. The user is the customer for the project, and unless the project is a research and development (R&D) project, the user pays for the project. The requirements directorate performs the acquisition on the behalf of the user. The requirements directorate starts the project by appointing a Project Director and a PM, and by creating a PMO. The Project Director has executive responsibility for the project. The PM has the day-to-day project management responsibility for the project. The PMO consists of the PM's supporting staff. Depending on the size of the project and the amount of work that will have to be done to design and manage the project, the PMO can consist of anything from just the PM to a fairly large staff. In a large project, the PM will usually delegate individual PMO staff to manage day-to-day activities in particular components of the project. For example, one PMO staff member could be assigned to look after HFE activities.

The project organization just described is used for ordinary acquisitions. If the acquisition is likely to require a significant R&D effort, a slightly different organization will be set up. Thus, if the acquisition involves new technical problems or untried technology, the acquisition responsibility for the project can be assigned to the Defence Research and Development Branch (DRDB). For the R&D Branch to be involved, the project has to not only include R&D problems, but also fit in with the R&D Branch's general R&D program priorities. If the project is set up as an R&D project, the R&D Branch pays for the project, and becomes a co-user along with the original user organization. For contractor HFE personnel, there is often no practical difference between an ordinary acquisition project and an R&D project, other than the difference in research emphasis.

The next subsections describe the project management tasks that get under way once the PMO has been organized.

3.2.1.2 Defining System Requirements from the Statement of Deficiencies

Once the requirements directorate creates a PMO for the acquisition, the PM's first task will be to take the user organization's statement of deficiencies and translate it into a statement of system requirements. Translating the statement of deficiencies into a statement of requirements can be hard to do. This is because the nature of command and control activities makes it difficult to formulate objective and quantitative statements of deficiencies.

The difficulty of formulating a statement of deficiencies for a CCIS can be seen by returning to the earlier CCIS-hardware system comparison. An ideal statement of a deficiency meets two criteria.

1 It implicitly defines the technical requirements that a new system would have to meet to remedy the present system's deficiency.

2 It defines a specific performance criterion that can be used to show whether or not the new system meets the requirement.

In a military hardware system, it is generally easy to compose statements of deficiencies that meet these two criteria. Using an armoured vehicle as an example, it would be relatively easy to determine if the maximum speed of the existing vehicle is too slow, and at the same time, specify the maximum vehicle speed needed to meet expected tactical demands. This expression of the deficiency in a current vehicle would give the PM a clear statement of a requirement, and later on, give the contractor an explicit design goal, and finally, give everyone connected with the project an objective criterion for deciding whether or not the ultimate system design overcame the original system deficiency.

In a CCIS, good statements of deficiencies are much more difficult to produce. First, just how the performance of the CCIS affects the performance of the command and control system as a whole is not always obvious. And typically, the user organization has few objective quantitative data on either the performance of the existing command and control system, or the performance of the existing CCIS. This leads the user to state deficiencies with generalities like, 'our planning process is too slow,' or, 'messages take too long to get through the system.' Such statements give the PM no basis for defining an explicit system requirement, nor do they give the system developer specific design goals or criteria for determining if a new CCIS's design will satisfy the user's needs.

The CCIS HFE Requirements User Manual puts considerable emphasis on the need to construct explicit statements of deficiencies. The User Manual (5.2) describes procedures that can be used to evaluate a user's statement of deficiencies, and steps to take to make the statement of deficiencies adequate for defining system requirements. Section 4 (page 27) in this CCIS HFE Design Guidelines describes how to apply the CCIS HFE Requirements and different HFE analysis methods to developing and refining a CCIS statement of deficiencies.

Once the PM has an adequate statement of deficiencies, the statement can be translated into a statement of system requirements, and the PM can go on to the next step in the project.

3.2.1.3 Investigate Options

As soon as the current system's deficiencies and new system's requirements have been identified, the PM begins to explore options for satisfying the requirements. Exploring options can consist of consulting with other military organizations, searching literature, talking to vendors and contractors, and generally exploring any courses of action that might lead to a potential option for meeting the system requirements. As said earlier, all the options that the PM is likely to find fall into three categories: 1) buy an off-the-shelf system, 2) develop a new system, or 3) conduct an exploratory research and development project.

There are several HFE considerations in investigating acquisition options. The CCIS HFE Requirements User Manual (5.3.1) advises the PM about these HFE considerations. The following points summarize this advice.

1 In comparing options, the option that best supports users' command and control tasks will generally also have the best HFE characteristics.

2 The results of HFE analyses of the user's command and control operations provide a good foundation for evaluating the merits of different options.

3 It may be useful to consult HFE literature or an HFE specialist when HFE questions come up in the course of evaluating options.

4 The contents of the CCIS HFE Requirements can be tailored to provide the basis for an evaluation of an off-the-shelf acquisition option.

Section 5 in this CCIS HFE Design Guidelines will describe how to adapt the CCIS HFE Requirements to investigating and evaluating options for meeting the acquisition system requirements. Section 5 will also describe how to adapt the CCIS HFE Requirements to evaluating an off-the-shelf system.

3.2.1.4 Designing the CCIS Development Project

As soon as an acquisition option is chosen, the PM begins designing the project. Designing the project includes the following tasks.

1 Create a master development plan that, if necessary, breaks development into a sequence of sub-projects.

2 Identify a system development approach and specify the technical standards and guidelines that will serve as authorities for conducting the project.

3 Construct a project budget estimate broken down by project cost components and project schedule.

4 Construct a project schedule defining timings of different phases of the project including timing of phases of the acquisition project.

5 Develop an SOW and RFP for soliciting proposals from potential contractors.

6 Conduct the contracting process leading to selection of a contractor.

In a project of any significant size or complexity, it takes considerable effort, and usually a number of iterations to complete the first five tasks to the point where all the parties are satisfied and the RFP can be sent out to potential contractors. To complete the tasks successfully, the PM has to predict in a fair amount of detail what the contractor would do to perform the project, and estimate in similar detail what it would cost to do the work, and finally, how long it would take to do different phases of the project.

The CCIS HFE Requirements User Manual (5.3.2) advises the PM on how to handle the HFE aspects of project design. The next four subsections summarize this advice.

3.2.1.4.1 Acquisition Project Master Plan

The master plan for a CCIS project delineates how the overall project will be broken up into sub-projects. Acquisition projects for large or complex systems, especially if they involve research and development, are usually broken up into a sequence of sub-projects. In large or complex systems projects, it is difficult to foresee exactly how each technical hurdle in the project will be overcome, and similarly, difficult to estimate the details of the project cost and schedule. Breaking the acquisition project into sub-projects makes it possible to minimize the risks inherent in committing to one grand project that might or not succeed, or that might require major revisions to deal with unexpected problems. By having the project divided into sub-projects, it is possible to accomplish each project hurdle without a large budget commitment, and at the same time, keep options open for how to approach succeeding steps in the project.

There is no standard way to break a project down into sub-projects. However, all approaches have the goal of minimizing risk by doing the project in manageable steps so that each technical problem is solved before the next one comes along. The following development steps are representative of how a large system development project can be broken down into smaller sub-projects [20].

1 Conduct a Requirements Study to clarify or elaborate the user's initial statement of the current system's deficiencies and the requirements to be met by the new system.

2 Do an Initial Development project to produce a system concept design or preliminary design that can be used to plan further development, and to estimate the chances of eventually being able to satisfy the system requirements.

3 Do an Advanced Development project to produce a detailed design and a working system model that can perform the functions required in the operational system.

4 Do an Engineering Development project to produce and implement the operational system.

This sequence of projects progressively takes development from the user's initial statement requirements to increasingly concrete design renditions that end with implementation of an operational CCIS. If the project is complex enough, the separate projects just listed could be broken up into still smaller sub-projects. Conversely, in a small system with no technical

uncertainties to resolve, the projects could be combined to form phases in a single project. However, no matter what the size of the project, the master plan for the acquisition has to chart a course that shows how the project will get from the user's initial statement of deficiencies to a practical operational system that meets the user's needs.

Each of the different types of sub-projects in a CCIS development has a different set of system engineering and HFE activities. In a requirements study, the results of HFE analyses provide a basis for refining and quantifying those statements of deficiencies that concern the performance of command and control personnel. In an initial development project, HFE contributes to function allocation in the initial system design, and to estimates of whether or not the design can meet the command and control personnel performance requirements specified for the new system. During engineering development, HFE activities contribute to system design, design evaluation, and design testing. In engineering development, the HFE emphasis turns to implementation concerns such as personnel selection, job requirements or job descriptions, and training.

Table 1 summarizes the system engineering and HFE activities that could be carried out in the different stages of a CCIS development. The CCIS HFE Requirements User Manual (5.3.2.2) gives the PM advice on how to adapt the CCIS HFE Requirements for each of the four types of development project. The PM can specify any necessary tailoring in the RFP, or simply invoke the CCIS HFE Requirements and make the contractor responsible for producing a suitable tailoring scheme. Section 5 in this CCIS HFE Design Guidelines describes the considerations that HFE personnel should take into account in tailoring the CCIS HFE Requirements to a particular project.

Once the PM and the other parties to the acquisition have decided on a master plan containing one or more separate projects, the PM can get on with planning the first project. The following sections describe what goes into designing an individual project. Although the approach to project design will vary somewhat depending on the stage of the acquisition's development, the basic tasks of specifying the system engineering and project management methods, and working out a budget and schedule are the same for all projects.

3.2.1.4.2 Define the System Development Standard and HFE Requirements

A system development standard specifies the technical approach to be used in the project, the main phases and milestones in the project, the planning and management documentation that the contractor will provide during the work, and the program of reviews that the acquisition authority will conduct to evaluate the progress of development. In general, a system development standard specifies the technical and project management requirements for the project's system development program in the same way that the CCIS HFE Requirements specifies technical and management requirements for the project's HFE program. The primary function of the system development standard is to provide the acquisition authority and the contractor with a detailed set of ground rules for conducting the project.

Table 1 Summary of system engineering and HFE activities in different types of CCIS development projects.

Project Type	System Engineering Activities	HFE Activities
Requirements Study	- do mission analysis and preliminary system analysis - produce statement of requirements	- do mission, function and high level task analysis - identify measures of performance - identify user support needs
Initial Development	- do detailed function and system analyses - identify concept design or concept design options - develop design representations such as mockups or prototypes - conduct concept design tests with users - produce a design and estimate likelihood of satisfying requirements	- do or expand detailed function and task analyses to support design - keep task analyses current with design - ensure user support functions and interface design conform to HFE design criteria - analyse design impacts on user workloads and user performance - conduct tests of designs and integrated system trials with samples of users
Advanced Development	- produce detailed system design requirements - do detailed design and conduct design tests - produce the development model - conduct trials using realistic mission scenarios - document design and the results of tests	- do detailed mission, function and task analysis - identify measures of performance and effectiveness - do function allocation and analyse function allocation trade-offs - influence design of user support functions and system-user interface - conduct design tests with users and evaluate design
Engineering Development	- produce final design - build and integrate system components - produce operating manuals, training, etc. - conduct acceptance trials - implement system and provide system life cycle support program	- up-date HFE analyses to reflect final system design - confirm design's compliance with HFE criteria - contribute to production of user manuals, procedures manuals, training, and personnel planning - conduct acceptance trials with users doing mission scenarios

There are a wide variety of system development approaches and technical standards or guidelines that a PM can choose from for managing a CCIS project. Until recently, DoD-STD-2167A [21], and to some extent its successor, MIL-STD-498 [22], were widely used to specify the system development approach. More recently, a private industry proprietary approach, METHOD/1 [23], has begun to replace DoD-STD-2167A. At the time of writing, there is no reliable way of telling what system development approach might be applied to any particular CCIS project.

Because none of the available system development approaches contain any significant provisions for HFE work, the CCIS HFE Requirements needed to be a standalone package that could be combined with any engineering development standard. To fulfill this need, the CCIS HFE Requirements contains, along with its HFE design guidelines, a complete set of devices (the CCIS HFE Requirements DIDs) for planning and managing the HFE work. Having these HFE project

management tools lets the PM specify HFE requirements for a project simply by invoking the CCIS HFE Requirements in the project RFP: for example, by putting the following statement in the RFP.

> `The system development plan shall include a human factors engineering plan (HFE Plan) conducted in accordance with *Guidelines for Human Factors Engineering Requirements for Canadian Forces Command and Control Information Systems* (CCIS HFE Requirements)'

The primary reason for making it possible to specify an HFE plan just by invoking the CCIS HFE Requirements is to make it easy for a PM to include HFE requirements in the project requirements. However, although the PM can specify HFE requirements just by invoking the CCIS HFE Requirements, it may be necessary to tailor the application of the HFE requirements to the overall system development standard chosen for the project. This tailoring is necessary for two reasons: 1) very important, to ensure that the HFE plan is integrated with the overall system development plan, and 2) to ensure that the HFE Plan includes only those parts of the CCIS HFE Requirements that are relevant to the scope of the project.

The CCIS HFE Requirements User Manual (Section 5.3.2.2) provides a detailed discussion of how to tailor the CCIS HFE Requirements to different system development approaches and different project management methods. The User Manual also contains a set of model statements specifying the CCIS HFE Requirements tailoring for the project's development scope, project development approach and project management method. If the PM chooses to, the PM can specify how the application of the CCIS HFE Requirements is to be tailored to the project.

If the PM does not specify the CCIS HFE Requirements tailoring, the contractor's manager responsible for HFE will have to devise a tailoring scheme and have it approved by the PM before the project starts. Normally, this tailoring scheme will be put in the contractor's proposal. In any event, the CCIS HFE Requirements User Manual will provide detailed advice on how to do this tailoring. Tailoring the CCIS HFE Requirements to the project's scope, and to the project's technical and project management requirements, is fundamental to ensuring that the HFE Plan has a significant impact on the system design. Therefore, the CCIS HFE Requirements User Manual (Section 5.3.2.2) should be treated as required reading for anyone preparing an HFE Plan that tailors the CCIS HFE Requirements to a CCIS project.

3.2.1.4.3 Account for the HFE Plan in the Project Budget

As part of designing the project, the PM has to estimate the project budget. The budget includes not only the direct cost of the development project, but also all the costs associated with managing the project. The PMO costs may include such things as the cost of the PMO itself, travel for review meetings, and personnel and facilities costs for system trials.

The direct cost of the project is normally the amount that the PM expects to have to pay a contractor to do the project. To estimate this cost, the PM has to do a fairly detailed estimate of

a contractor's prices for different components of the project. This includes estimating the cost of the HFE Plan. For estimating the costs of most of the project's system engineering components, the PM can rely on a variety of resources: the acquisition authority's collective experience from other projects for example. This will not be true for the HFE Plan. Projects with significant and well-integrated HFE plans are rare, so there is not much costing experience to go on, even for HFE specialists. Because PMs and PMO personnel typically have limited HFE backgrounds, they are at an additional disadvantage in trying to estimate an HFE budget.

Presuming these costing difficulties, the CCIS HFE Requirements User Manual (5.3.2.5) advises the PM on how to estimate the cost of the HFE Plan by giving two alternative budgeting approaches:

1 assume that the cost of the HFE Plan will be between five and 10 percent of the project budget, or

2 get an HFE specialist to work out a budget for the HFE Plan.

The first approach gives the PM a guideline that can be used without having to know a great deal about the contents of the HFE work. The second approach gives the PM a means of getting a budget estimate based on a concrete analysis of the project's contents.

The first approach, using a percentage factor for estimating the HFE budget, is not an option for the contractor. Section 5 in this CCIS HFE Design Guidelines advises on how to estimate an HFE plan budget.

3.2.1.4.4 Account for the HFE Plan in the Project Schedule

Project design includes working out a schedule for the project. The CCIS HFE Requirements User Manual (5.3.2.6) advises the PM on how to take account of the HFE Plan in the project schedule. With one exception, all the activities in the HFE Plan can be done in parallel with other system engineering activities. The exception is any HFE system design test whose results are needed for subsequent design work. In most development projects, there need be only one occasion when other system engineering work may be interrupted by a design test. This is usually near the end of the project when the entire system design has to be tested. Before this point, design testing can usually be done on subsystems. In the case of subsystem testing, the contractor should be able to schedule work so that while one subsystem is being tested, work can continue on other subsystems. Consequently, provided that integrated system tests are taken into account, the overall project schedule will be a mainly a function of the non-HFE system engineering activities.

3.2.1.4.5 Engage a Contractor

Once the system engineering and project management standards have been defined, and a project budget and schedule have been worked out, the acquisition authority can issue an RFP, and hire a contractor to do the work. The appointment of a contractor is a turning point in the PM's

activities. It is the point where the PM shifts from planning the project to managing the execution of the project. The next section describes the project management phase of the project.

3.2.2 Managing the CCIS Development Project

For the PM, managing the execution of the project consists of three general tasks: 1) monitoring the progress of the contractor's work, 2) adjusting the project plan as necessary to deal with unanticipated problems, and 3) managing user participation in project activities. Performing these tasks is fairly straightforward when the project has been well designed to start with, and the contractor is competent to do the work.

The CCIS HFE Requirements User Manual (5.4) advises the PM on managing the execution of the HFE component of the project. The following paragraphs summarize the tasks the PM does to manage the project in general, and the HFE work in particular.

1 *Oversee the launch of the project plan.* The PM will ensure that each component of the contractor's work, including the HFE Plan, gets under way as planned or proposed, and that there are no loose ends that need to be cleared up before work can start. For the HFE Plan, clearing up loose ends may include making adjustments to the plan's budget or schedule, or clarifying the tailoring provisions of the CCIS HFE Requirements.

2 *Monitor and control the project budget.* The PM will monitor the expenditure of each component of the project budget to ensure that total expenditure will stay within the budget for the project. This may lead to adjusting the budgets for different project components from time to time.

3 *Monitor and control the project schedule.* The PM monitors the progress of each project component against the project schedule. Since almost all the HFE Plan activities are tied to the timings of other system development activities, delays in the HFE schedule will most likely come from delays in other system development activities rather than delays in completing HFE activities.

4 *Manage user participation in project activities.* User participation will be needed throughout the project for system analysis and HFE analysis activities, for conducting system design tests, and for system design reviews. The PM is responsible for organizing user participation in system analysis and testing activities, and if required, getting access to any user facilities that might be required.

5 *Review and approve the contractor's reports and deliverables.* The PM reviews and approves the reports and other deliverables defined in the project management specifications for the project. For HFE work, the reports and deliverables will be those described in the CCIS HFE Requirements DIDs that were specified for the project.

6 *Maintain a progress review process with users and stakeholders.* The PM is responsible for ensuring that users and other interested parties are kept informed about the progress of the project. Formal reviews specified in the project requirements are the primary vehicles for

maintaining the review process. Reports specified by HFE DIDs, particularly the design test reports, will be the primary vehicles for conveying the results of HFE work.

All of the tasks that the PM does to monitor the HFE work are the same as the tasks that the PM also does to monitor other components of the project, and does to manage the project as a whole. In general, the PM's tasks during the execution phase of a project aim at making sure that the project is carried out as intended in the project design. If the project strays from the course set out in the project design, the PM must find a way to get the project back on its course. A project is off-course when milestone objectives are not going to be met on time, or not met within budget, or not met at all. When a project is off-course by a modest degree, the PM can usually correct the situation by making modest schedule adjustments in some project tasks, or by reallocating budget among different project components - usually without delaying the planned project completion time or going over the total project budget. When a project starts going badly off-course, it is typically because important system requirements or important project goals cannot be met - usually because of technical problems that could not be solved, and occasionally because of mismanagement of project activities.

The amount of autonomy the PM has in project management decision making varies with the nature of the project and the nature of the project organization. Generally, the PM has sufficient authority to exercise day-to-day control and make project plan adjustments to deal with difficulties that do not threaten the achievement of the project's objectives. When a serious problem emerges, and the user organization or other authorities have to be involved in dealing with the problem, the PM coordinates the decision making process. This coordination will likely include developing options for dealing with the problem, much the same way acquisition options were developed during the project design phase.

The HFE activities in a CCIS development project are fairly immune from becoming significant factors in a project failing to meet its goals. Provided that the contractor's HFE personnel are reasonably competent, and they follow technically sound HFE practices, there are no HFE activities likely to subject the project to an undue risk of failing to meet user requirements. On the other hand, if the HFE Plan is not carefully integrated with the overall system development plan so that the system design reflects the results of the HFE work, the HFE work can be a major indirect cause of a CCIS failing to meet its performance objectives. Although HFE personnel cannot force integration of the HFE results into the system design, the HFE personnel have a major responsibility for ensuring that their work is structured so that integration is not only possible but convenient.

3.2.3 Summary of CCIS Project Management

Every project contains an underlying project management effort that begins well before a contractor starts work on the project. The following summarizes the main steps in the CCIS project management process.

1 Project design.

1.1 The user organization identifies deficiencies in its present CCIS and starts the acquisition process by creating a statement of the existing system's deficiencies.

1.2 The acquisition authority sets up a PMO, and the PM develops a statement of system requirements based on the user's statement of deficiencies.

1.3 The PM then does research to identify the best acquisition option for procuring a CCIS that will meet the user's requirements.

1.4 The PM designs the acquisition project by choosing a project management and system development approach, creating a project budget and schedule, and coordinating the selection of a contractor to do the development work.

2 Project execution.

2.1 The PM ensures that the contractor's work gets under way with everyone having a common understanding of the project's goals.

2.2 The PM monitors project management reports to track the contractor's progress toward deliverables relative to the project's budget and schedule.

2.3 When the project runs into unexpected difficulties, the PM coordinates decision making on how to adjust the project plan to respond to the difficulties.

2.4 Throughout project execution, the PM coordinates user participation in such things as system analysis and HFE analysis activities, and user participation in system design tests.

3.3 Summary of HFE Background Issues in CCIS Development

The following points summarize the issues that affect the application of HFE methods to CCIS development.

1 *HFE Objectives in CCIS development.* The objectives of HFE activities in CCIS development are the same as the objectives of HFE in any other application: to create a system that lets users perform their work safely and efficiently, and so provides the user organization with the most cost-effective use of its personnel resources.

2 *The nature of command and control.* A command and control system is made up of three components: 1) the commanders and staff who plan, direct and monitor operations, 2) the doctrine and procedures that commanders and staff follow in performing command and control activities, and 3) the facilities and equipment that support command and control operations. A CCIS is part of this third component. The definition of command and control implies two basic objectives to be met in developing a CCIS: 1) the new CCIS must improve the commander's effectiveness by improving the ability of the commander and the commander's staff to plan, direct or monitor operations, and 2) the new CCIS must preserve the integrity of the doctrine and procedures that represent the principle of command and control and the logical structure of the command and control system. The nature of command and control, and its underlying principles mean that the ultimate goal of a new

CCIS is to improve the performance of its users. It is for this reason that HFE has an important role in CCIS development.

3 *The difference between CCIS and hardware systems.* The entire function of a CCIS is to support the human tasks of the commander and staff who are its users. A new CCIS must do this while preserving the user organization's principles of command and control. In contrast, in a military hardware system, the humans who operate the system are essentially subsystem elements whose functions are determined by the overall system design. This difference between CCISs and a hardware system requires different HFE system develop approaches. In hardware system development, HFE work begins after the initial system design and flows from the functions assigned to the system's human operators. In CCIS development, to ensure understanding of the user's command and control functions, HFE work must begin before the initial system design. And rather than HFE work following from the system design, much of the system design must follow from the results of the HFE work. As a consequence, CCIS HFE activities need to be more closely integrated with other system engineering activities than is typical in hardware system development. And for this integration to be effective, HFE personnel in CCIS development have to be more aware of system development project management procedures than they typically need to in hardware system development.

4 *The CCIS project management process.* The CCIS acquisition process begins with the user's statement describing the deficiencies in the existing CCIS. The acquisition authority responds by setting up a PMO with a project director, PM and PMO staff. The PM manages the acquisition process by doing four tasks.

4.1 Translate the user's statement of deficiencies into a statement of system requirements, and then find the best option for satisfying the requirements.

4.2 Design the acquisition project by defining the system engineering and project management method, and the project budget and schedule.

4.3 After a contractor is hired to do the acquisition project, manage the completion of the acquisition by monitoring the contractor's progress against the project budget and schedule.

4.4 As part of the overall acquisition process, specify the HFE requirements for the system, take account of the HFE work in the project design, and monitor the progress of HFE work during the execution of the project.

5 A major factor in the success of the HFE work will be the degree to which all HFE personnel associated with a CCIS project take pains to ensure that the HFE work integrates smoothly with the project management process and with other system engineering activities.

This section has given an overview of both the HFE technical and HFE project management issues in CCIS development. The next section will describe details of the HFE technical methods that may be used in applying the CCIS HFE Requirements to a CCIS development project. A subsequent section will provide advice on how to use the project management devices in the CCIS HFE Requirements to plan and manage the HFE work in CCIS development.

4 HFE METHODS IN CCIS DEVELOPMENT

This section describes HFE methods that are applicable to CCIS development, and the relationships between these methods and different parts of the CCIS HFE Requirements. The material is intended for qualified HFE specialists. Nothing in this section is meant to limit the HFE methods that an HFE specialist judges to be suitable in a specific situation. In general, the contents of this section concentrate on the *process* of applying HFE to CCIS development rather than on defining *design rules*. The design guidelines references in Section 2, along with other HFE references, should provide adequate sources for looking up specific design advice. Rather than suggesting what constitutes good HFE design, the advice in this section aims at helping the HFE specialist to create and carry out an effective HFE plan.

The contents of this section also serve as a counterpart to CCIS HFE Requirements User Manual (4) which aims at giving the PM and PMO staff a basic appreciation of HFE activities in CCIS development, and particularly, a basis for evaluating the technical quality of the contractor's HFE Plan and the quality of the HFE work performed by the contractor. CCIS HFE Requirements User Manual (4) will be helpful to the HFE specialist for getting an understanding of what is expected by the acquisition authority PM. However, it is not a substitute for direct consultation between PMO staff and HFE personnel to ensure that there is mutual understanding of the HFE requirements at every stage of the project.

The material in this section will be organized around the structure of the detailed requirements the CCIS HFE Requirements (5). CCIS HFE Requirements (5) is divided into three main subsections, HFE analysis requirements, requirements for HFE system design, and requirements for HFE in design testing. Correspondingly, this section on HFE methods will have three main subsections: 1) HFE analysis methods, 2) HFE design methods, and 3) HFE testing methods. In addition, the coverage of each topic will include a table of cross-references to CCIS HFE Requirements.

The organization of the CCIS HFE Requirements corresponds to the sequence in which HFE analysis, design and testing activities would occur in a conventional CCIS development project. However, in a CCIS project of any substantial size, the project will likely be broken into sub-projects like the Requirements Study, Initial Design, Advanced Development and Engineering development sequence described earlier in Section 3.2.1.4.1 and Table 1. Besides the main stages being broken into sub-projects, each will likely contain iterations of a cycle of analysis-design-prototyping-test and evaluation (see Section 4.3.1.5, page 141). To help relate the HFE methods to this type of phased development, each of the following HFE methods subsections will describe how the different HFE methods apply to different phases of development.

Each of the following subsections will also describe how the CCIS HFE Requirements DIDs can be used to plan and report on HFE activities. Although the DIDs are primarily HFE project management devices, they are also constructed to serve as templates for designing the technical content of HFE activities, and for reporting on the results of these activities. Used as templates for the planning HFE work, the DIDs will minimize the general problem solving effort needed to

work out plans for HFE activities. Used as templates for reporting, the DIDs will minimize the overhead effort needed to document the HFE work.

Based on the organizing principles just outlined, the remainder of this section will contain four main subsections.

1 *Overview of the HFE process in a CCIS development project* will describe a scenario outlining the possible HFE activities in a CCIS development project, starting when the PM develops a statement of requirements, through system design, and ending with implementation.

2 *HFE analysis methods in CCIS development* describes the HFE analysis methods that are most applicable to different phases of CCIS development. The analysis methods subsection covers mission analysis, function analysis, task analysis, and other special analyses such as operational sequence analysis that can be useful for analysing command and control operations.

3 *HFE in CCIS design* describes the HFE activities that contribute to system design. These activities include providing advice on design, reviewing the HFE qualities of design, and keeping function and task analysis information current with design.

4 *HFE in CCIS design testing* describes methods for testing system designs. The methods cover both testing for the adequacy of a design's HFE characteristics, and testing whether or not a design meets user performance requirements.

4.1 Overview of HFE in CCIS Development

The following overview briefly describes the tasks that an HFE specialist could be asked to do during a CCIS development process. Some of these are tasks to help the PM design the acquisition project. Others form part of the CCIS system engineering process.

4.1.1 Formulate Initial Statement of Requirements

Earlier, Section 3.2.1.2 described the difficulty that a PM is likely to have in deriving a statement of requirements from a statement of a CCIS's deficiencies. Most of the difficulty comes from trying to derive objective and quantitative requirements from essentially subjective statements of the existing CCIS's deficiencies - deficiencies that are typically expressed indirectly as deficiencies in the performance of command and control personnel. An HFE specialist can help in formulating a statement of requirements in several ways. At the simplest level, an HFE specialist can advise on how to define objective and quantitative human performance measures linked to the CCIS's performance. If more intensive effort is needed, the specialist can conduct preliminary HFE analyses of the existing command and control operations to identify any CCIS deficiencies that hamper the users' performance of their command and control tasks.

4.1.2 Account for HFE Work in the Project Budget and Schedule

If the PM wants to do a direct estimate of the project HFE budget, rather than use the five to 10 percent budget factor suggested in the CCIS HFE Requirements User Manual (5.3.2.5.2), the PM could engage an HFE specialist to sketch out the project HFE plan and use the plan to estimate the cost of the HFE work. The same HFE plan could also be used to adjust the project schedule to take account of HFE activities.

4.1.3 Develop an HFE Plan for Inclusion in Bidding Contractor's Proposal

Even if the RFP does not specify doing so, the most obvious way to write a proposal for the project HFE work is to use the CCIS HFE Requirements HFE Plan DID as the basis for the HFE proposal. This would include working out the preliminary HFE Plan and helping to estimate its cost and project schedule impacts.

4.1.4 Help the PM Select a Contractor

An HFE specialist can help the PM in contractor selection by reviewing the HFE sections of proposals and estimating the relative merits of the different proposals. The CCIS HFE Requirements User Manual (5.3.3) advises the PM on how to evaluate the HFE components of a proposal. An HFE specialist can use the same section as the basis for evaluating HFE proposals.

4.1.5 Act as Independent Verification and Validation (IV&V) Agent

Acquisition contracts over a certain size require an IV&V agent independently to evaluate the technical adequacy of the project contractor's work. An HFE specialist can provide IV&V services for the HFE components of a contractor's work. The CCIS HFE Requirements does not include any requirement for an HFE IV&V agent, and consequently, does not specify anything about how an IV&V agent should evaluate a contractor's HFE work[6]. However, the CCIS HFE Requirements does provide the obvious basis for conducting IV&V reviews of HFE work.

4.1.6 Conduct HFE Analyses and Tests for Detailed System Requirements

These analysis and test activities are part of the first phase of a CCIS development project. The CCIS HFE Requirements (5.1) contains the requirements for these activities. The activities are mostly ones that take the project to the creation of a preliminary system design, or at the least, to detailed design specifications. The HFE analyses include mission, function and task analyses of the existing command and control system. Results of these analyses will help to identify

[6] Since IV&V activities are outside the scope of the CCIS HFE Requirements, there will be no discussion of IV&V in this CCIS HFE Design Guidelines other than in this paragraph.

requirements for the new system, particularly the allocations of command and control functions between user and CCIS. If this project phase includes developing a preliminary system design, HFE analyses of how users would operate the new system could be used to evaluate the merits of preliminary design options. HFE activity in early design work may also include conducting HFE experiments to test design assumptions about the user capabilities demanded by the system design: for example, whether or not users could handle the workloads[7] imposed by certain design features.

4.1.7 Perform HFE Tasks in System Design and Design Testing

The HFE tasks in system design include four activities: 1) providing design advice to maximize the HFE qualities of the design, 2) reviewing design options against HFE criteria, 3) keeping the HFE analysis documentation current with the system design, and 4) conducting tests of design options. CCIS HFE Requirements (5.2) contains the requirements for the first three activities: participating in design and keeping HFE analyses up to date. The CCIS HFE Requirements (5.3) contains the requirements for the fourth activity, design testing.

Keeping the HFE analyses current with system design is fundamental to providing design advice and evaluating the HFE quality of design options. Although one can give a good deal of design direction by applying general HFE design rules and principles, the only truly effective way to provide HFE design advice is from having a thorough knowledge of users' tasks.

Up-to-date HFE analysis data are also fundamental to doing useful design testing. The HFE analysis data are the foundation for identifying measures of user performance for design tests. Tests that show how well users can perform system operating tasks are the only way to decide if a system design will support eventually users' command and control tasks. Without user performance measures, design tests degenerate into system demonstration exercises in which test participants are asked how they like the system.

Design tests range in scope from small tests of separate subsystems to large-scale system acceptance trials in which users use a prototype of the complete system design in simulated command and control operations. Design tests can begin almost as soon as design gets under way.

4.1.8 Participate in Implementation

HFE contributes to implementation by helping in the development of user manuals, user training, documenting user operating procedures, and helping to define personnel selection requirements. Again, the HFE analysis data provide the foundation for this contribution. HFE implementation activities may be done as part of the CCIS development project, or as separate activities after the system has been delivered. Whether or not there is direct HFE participation in implementation,

[7] Throughout this document, workload means mental workload, not physical workload.

documentation of the HFE work done during the development project remains a valuable resource for all matters concerning personnel issues throughout the life cycle of the system.

4.1.9 Summary of HFE in Development

There are many tasks that an HFE specialist may be asked to do during a development project. Some tasks contribute to designing the project. Others contribute to the system development process. Others contribute to implementing the system in operations. Although the intended scope of the CCIS HFE Requirements is limited to the system development activities, the Requirements can still be adapted to guide almost any HFE work done from the time the user creates a statement of deficiencies to the time when the system reaches the end of its life cycle.

Obviously, conflict of interest considerations would prevent any one HFE specialist from participating in all the HFE tasks that might be done during an acquisition project. For example, the same specialist could not help the PM to create an RFP and conduct a contractor selection process, and then turn up on a bidding contractor's HFE team. In large projects broken into several sub-projects, different contractors with different HFE teams may be hired to do the different sub-projects. Finally, in large projects that span several years, chances are that many people who worked on the project at the beginning will not be the same people working on the project at the end. Because different people may work on a project at different times, the documentation of the HFE work is critical to maintaining development continuity and preventing duplication of effort.

4.2 HFE Analysis Methods in CCIS Development

This section describes the HFE analysis methods that are applicable to CCIS development. The section is divided into the following parts:

1 a note on the nomenclature used to identify the different types of HFE analyses;

2 a discussion of the relationship between HFE analysis activities and other system analysis activities;

3 descriptions of the four basic types of HFE analysis applicable to CCIS development:

- 3.1 mission analysis;
- 3.2 function analysis;
- 3.3 task analysis, and
- 3.4 analysis using computer simulation; and
- 3.5 development of measures of user performance and effectiveness from HFE analysis data.

This last subsection discusses the need to identify measures of user performance and effectiveness from the HFE analysis results.

4.2.1 Note on HFE Analysis Nomenclature

An HFE analysis is any analysis that provides a systematic description and understanding of a human activity. Many analysis methods can be applied to analysing human performance. Beevis *et al.* [18] describe more than 20 types of analyses. In the following, the HFE analysis methods that are most useful in CCIS development will be grouped into four categories: 1) mission analysis, 2) function analysis, 3) task analysis, and 4) analysis by system simulation. These categories include many of the analyses described by Beevis *et al.* The ones described by Beevis *et al.* that will not be mentioned here are ones that would rarely be useful in CCIS development.

The division of analyses into the four categories, mission analysis, function analysis, task analysis, and operational sequence analysis, is somewhat arbitrary. This is because there is no firm agreement in the literature on how to classify the different analyses. For example, Luczak, in the Handbook of Human Factors and Ergonomics includes most four groups of analyses in one chapter titled Task Analysis [24]. Beevis *et al.* [18], on the other hand, classify many of the analyses that Luczak calls task analyses as function analyses, and classify what will here be called operational sequence analyses as a task analysis and workload measurement methods. Contrary to Beevis *et al.*, Laughery and Laughery [25] in the Handbook of Human Factors treat operational sequence methods as function analysis methods. The different classification schemes used by different authors can be confusing to one who is looking up literature on a particular type of analysis. However, the confusion about classification should not interfere with learning or applying any particular analysis. While different authors choose different headings for discussing different analyses, they seldom disagree on the purpose or nature of any particular analysis method.

The division of analyses into the categories used in the following sections is the division judged most logical for CCIS applications, and judged most convenient for the reader. The analyses are described in roughly the sequence in which they would be applied in a development project. Their sequence also represents a progression from analysis of the overall system to analysis of individuals' tasks. Beyond these two principles, the organization of the material is essentially arbitrary.

Another source of terminology confusion comes from the fact that except possibly for task analysis, the HFE analysis methods described here have counterparts with the same names in other system engineering disciplines. In fact, many HFE analysis methods are system analysis methods applied to analysing human behaviour in a system. In the following, the HFE qualifier will be used only when needed to differentiate the HFE method from its equivalent in another discipline. Generally the equivalent analysis methods in different system engineering disciplines look at the same things. How each method differs across disciplines is mostly in the purpose of the analysis, the types of information extracted from the analysis, and the inferences made from the analysis. For example, an HFE mission analysis is mainly concerned with what the humans in a system do to accomplish a system mission. A mission analysis in another discipline might be primarily concerned with what a certain element of system hardware does to accomplish a mission.

The fact that different disciplines use similar methods implies that it is often possible to get a lot of HFE analysis information from the results of other system analysis work. Conversely, HFE analysis results can often be used in other system analysis purposes. So, HFE analysis should not be done in isolation. It should be coordinated with other system analysis work to avoid duplication of effort and to ensure that, whenever possible, HFE efforts contribute to other system engineering efforts. Consequently, HFE analysis work must be documented not only with HFE users in mind, but also with other system engineering users in mind.

4.2.2 Scope of HFE Analyses

The substance of HFE analyses is always concerned with the human elements of the system. While to be meaningful, many analyses must include the system as a whole, both its human elements and its hardware and software elements, the primary purpose of an HFE analysis is to understand the roles that humans do play or could play in a system's functions. Therefore, when system engineering analyses are used as HFE analysis techniques, or the results of system engineering analyses are used for HFE purposes, the central task of the HFE practitioner is to draw inferences about human performance capabilities and how to design the system to maximise these capabilities.

The time scope of HFE analyses, as Section 4.1 suggested, is essentially from project conception to system implementation. Analyses of users' functions and tasks can be used to help identify and quantify predecessor system deficiencies, and a little later, to specify the requirements for a new system. During system development, the results of ongoing HFE analyses provide the basis for supporting system design and conducting test of the system design. At implementation and onwards, the HFE analysis results provide support for all the human resources management activities associated with operating a system: training, personnel selection, job evaluation, and so on.

Considering the content scope of HFE analyses, the methods described in the following sections concentrate first on how the methods apply to understanding human behaviour, specifically behaviour in command and control systems. Because of the time scope of HFE analyses, the methods are necessarily described without setting conditions for when they should be used during a project. Although some methods self-evidently apply at particular stages of system development, most of them apply throughout development. Therefore, the analyses should be thought of as a set of tools to draw on anytime during development when they would contribute to the progress of development.

The following subsections describe the HFE analysis methods, each in turn.

4.2.3 HFE Mission Analysis

This section will cover the following topics:

1 the CCIS HFE Requirements sections and DIDs covering mission analysis;

2 the nature and purpose of a mission analysis;

3 a general description of the command and control cycle with descriptions of its three main operational phases: planning, directing and monitoring;

4 descriptions of mission analysis methods for the different phases of the command and control cycle, and for different types of command and control systems;

5 procedures for gathering mission analysis data, and

6 a discussion of how the mission analysis results can be used in other HFE activities.

4.2.3.1 CCIS HFE Requirements Specifications for Mission Analysis

The following describes which sections of the CCIS HFE Requirements that specify HFE mission analysis activities. The CCIS HFE Requirements (5.1.1.1) specifies the HFE mission analysis. Documentation of the mission analysis is specified in the CCIS HFE Requirements DID titled Human Factors Engineering Analysis Report. Mission analysis is also referenced in the DIDs titled Human Factors Engineering System Requirements and Function Allocation Report, and Human Factors Engineering Test Report. Although not directly referenced in other DIDs, the results of the mission analysis will bear on the work covered in most of the DIDs.

The CCIS HFE Requirements does not specify what methods or techniques are to be used to do the mission analysis. However, some combination of the approaches described in the following subsections should satisfy the requirements specified in CCIS HFE Requirements (5.1.1.1).

4.2.3.2 Purpose and Nature of Mission Analysis

The mission analysis has two primary objectives:

1 provide an overview of what a system does, and

2 provide a foundation for subsequent HFE analysis, design and testing activities.

Beyond the objectives stated, the mission analysis should identify any constraints on the proposed system. Constraints include such things as limits on manning levels, skill and training limitations, and so on. These constraints can be identified implicitly by identifying the intended system users, or identified directly from requirements set out in any Statement of Requirement developed from new templates adopted by the Directorate of Business Change Management. The new templates include sections that address the human factors issues listed in Table 2.

The entries in Table 2 will not provide design solutions, but should identify major constraints and top-level requirements. Thus, much of the material needed to develop an HFE plan should be provided to the developer at the outset of a project[8].

[8] David Beevis, personal communication.

Table 2 Directorate of Business Change Management human factors issues.

- Missions and Scenarios	- User Acceptance
- Key Roles	- Survivability
- Key Tasks	- Maintainability
- User Characteristics	- Availability
- System Effectiveness	- Reliability
- Operability	- Safety and Health
- Crew Station and Interface Design	- Personnel and Training

A mission analysis describes the main events during a system's operational cycle. Although mission analysis has no standard definition, usually, the term comes from describing the events in a weapon system's basic operational cycle[9]. For example, a very elementary description of a bomber's mission might contain the following events: 1) plan the mission, 2) fly to target, 3) drop the weapons load, 4) fly back to base, and 5) debrief. Obviously, each of these mission phases will be made up of its own sequence of events. Depending on the detail needed, there is almost no limit to the degree to which a system's mission can be broken down into discrete events. In practice however, keeping the mission description at a simple level provides a comprehensible overview of the system mission. If a mission description gets into much more detail than one can recall from memory, the analysis may be too detailed. The other HFE analysis methods described later in this section are better suited to getting at the details of operations.

A mission analysis also implicitly describes the general objectives of the system: a bomber's objective is to deliver bombs to a target. The term mission statement usually means a formal statement of the system objectives - though the term is probably now more common in civilian non systems settings, than in military settings. Although a mission analysis should include an adequate description of the system objective, it does no harm to have a mission statement that describes the systems objectives separately.

4.2.3.3 The Three Phases of the Command and Control Mission

In command and control, a mission is broadly defined by the military operation that the command and control organization controls. A bomber doing a bombing mission, or a land force carrying out an attack on an objective, or a naval force conducting an antisubmarine patrol is each an example of an operation that might be controlled. Every military operation includes a command and control component to plan and control the operation. This component is usually called the command and control cycle.

As described earlier, the command and control cycle has three basic phases: 1) planning the operation, 2) directing the operation by issuing orders for the operation, and 3) monitoring the execution of the operation until the operation's completion. In continuous operations, usually at least two, sometimes more, command and control cycles are going on simultaneously. For example, while some command and control staff are monitoring the execution of one operation, other staff may be planning the next operation. Commanders must keep track of and participate in all planning and monitoring activities that are going on at any one time.

[9] See the section on mission analysis in [7] for a discussion of different definitions of mission analysis.

The following will show that the three phases of the command and control cycle contain distinctly different types of human activities. Human activities in planning are procedural tasks involving information manipulation, problem solving and decision making. User workloads in planning are fairly predictable and manageable. Each command and control staff member has a well defined set of tasks that can be done with a reasonably stable division of labour among tasks. The HFE problems in planning are mainly concerned with reducing unnecessary repetitive tasks, and maximizing the speed and quality of decision making processes.

The human activities in monitoring, while procedural in nature, are driven by sequences of randomly occurring outside events. This results in unpredictable swings in user workloads that may range from nothing to overwhelming. In addition, there is often insufficient information to maintain a complete and up-to-date picture of the situation being monitored. The HFE problems in monitoring activities are mainly concerned with workload management, sharing attention among competing tasks, and maintaining comprehension of a continually changing situation.

Because of the different HFE characteristics of different command and control phases, different mission analysis methods are usually needed for different command and control phases. The following subsections describe mission analysis methods for each phase separately. The methods will be described for a single headquarters[10] controlling a set of subordinate units. A subsection following the ones that describe the basic analysis methods will discuss considerations for dealing with a multilevel hierarchy of headquarters.

4.2.3.4 Mission Analysis of Planning

Command and control planning is a highly procedural activity. It starts with an order from a superior command level. This order typically defines the objective and concept of the operation, along with certain priorities and constraints. It is then up to the subordinate commanders and staff to work out a detailed plan for achieving the intent of the order, and then get the superior commander's approval of the plan. Working out a plan includes detailed manipulation of information, coordination among different personnel, and risky decision making - all within the confines of the organization's standard operating procedures and military doctrine.

Analysing the planning phase of the command and control cycle can usually be done from operating procedures documentation. All military organizations have documented procedures for doing planning tasks. Although the documentation may not be all in one place, nor in easily comprehended form, it should nevertheless be a straightforward job to develop a mission profile of the planning process. In addition all military organizations' planning processes have a fairly common process. Figure 2 illustrates the main features of this process.

[10] Throughout this HFE Methods section, the term headquarters includes any organizational element that exercises integrated command and control functions. Other names used for such elements include control centre, coordination centre, command post, and the like.

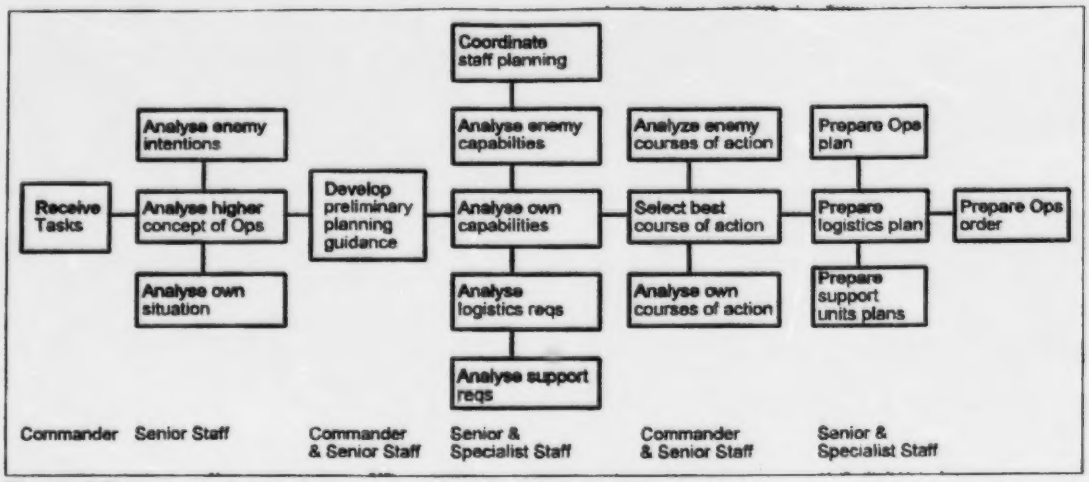

Figure 2 Flow chart of steps in a command and control planning process.

Figure 2 shows a flow chart of the steps in a generic command and control process planning process. The boxes in the flow chart show the different planning activities. The labels at the bottom of the chart show which members of the command and control organization carry out the different activities.

Planning begins with the commander receiving instructions or tasks from higher command. The commander then briefs senior staff on the tasks, and the senior staff do a preliminary analysis of the task requirements compared with anticipated enemy intentions, higher command's intentions, and the organization's own situation. Based on the results of these preliminary analyses, the commander issues planning guidance to be followed by different units in the command and control organization's staff. Based on the commander's planning guidance, the staff units do analyses covering such things as enemy capabilities, own capabilities, logistics requirements, and requirements for the organization's various support units.

Results of the preliminary staff analyses (often called staff estimates) provide the basis for identifying two sets of possible courses of action: one set of possible courses of action that the enemy might take, and one set of possible courses of action that the organization can take. The commander then selects a course of action and issues planning guidance to the staff. Staff then prepare detailed plans covering the combat operation, logistics support, and the tasks to be done by the organization's support units. When the commander approves these plans, they will be translated into an operations order. The operations order is usually a composite containing not only the orders to combat units, but also orders to all support units that will have roles in the operation.

The lines connecting the boxes in Figure 2 do not have any significance beyond showing the general sequence of activities, and linking related activities. Any planning process will have many feedback loops that are not shown in Figure 2. For example, the process would have set of loops

representing the commander's approval of outputs from each stage before the process goes to the next stage. An analysis of a particular system would have to include these loops, along with any others that might control the process.

The steps in the planning process in Figure 2 are nearly universal, whether they are done in the space of a few minutes by a platoon commander and his or her NCOs, or they are done over months by a multinational supreme headquarters with a staff of many hundreds. For the purposes of a mission analysis, the scheme shown in Figure 2 can be used as a template for constructing a flow chart showing the planning process in a specific command and control organization. Similarly, Figure 2 can be used as a checklist for deciding if an analysis is complete. In particular, any mission analysis should cover the following items:

1 the commander's receipt of an order or instructions from higher command,

2 the preliminary analysis of the higher command's instructions,

3 development of guidance to staff for initial planning,

4 the staff process for analysing combat task requirements and capabilities,

5 analysis of possible courses of action,

6 selection of a course of action and development of a concept of operations and outline plan,

7 preparation of detailed plans, and

8 translation of detailed plans into the operations orders.

A flow chart like the one in Figure 2 shows the sequential relationships among planning activities, but not the timing of activities. A Gantt chart, critical path chart, or PERT chart is a good way to represent these timings of activities[11]. Figure 3 shows a critical path chart for the activities shown in the flow chart in Figure 2. For the sake of generality, the time scale for the chart is in arbitrary units. In a large command and control organization, the actual time units would be in the order of hours. In a small organization, the units could be as small as minutes. A critical path chart shows two important sets of information, one of which is not available from the flow chart, and the other of which is not easy to see in a flow chart. The information that is not available in a flow chart is the length of time needed to complete each activity. The information that is not always easy to see in the flow chart is which activities are on the planning process's critical path.

Although producing at least a preliminary planning process critical path chart is usually easy once a flow chart has been worked out, getting the durations of the different activities may not be easy. For one thing, the command and control organization may not know the durations of the activities in any detail. In addition, the nature of the particular command and control system may result in there being considerable statistical variation in the durations of different activities. If either of

[11] A PERT chart allows probabilistic times for completing activities. A critical path chart is based on fixed times to complete activities.

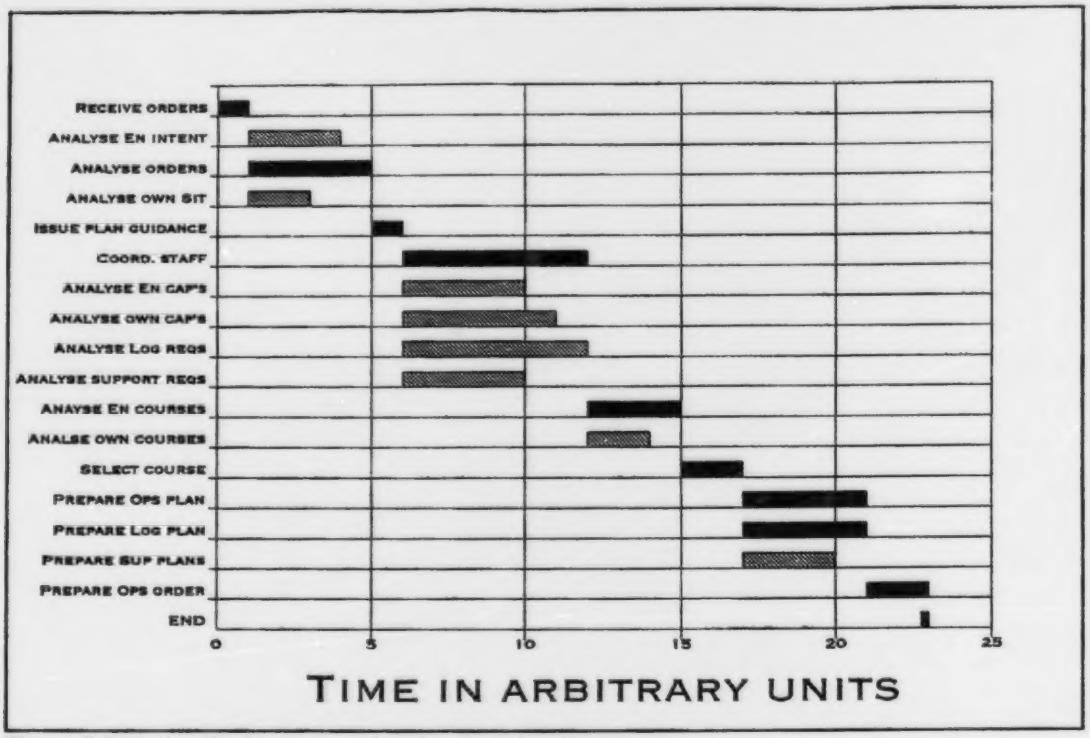

Figure 3 Critical path of a command and control planning process.

these factors prevents getting precise estimates of the activity durations for the mission analysis, getting estimates of the activity durations is best left for later detailed analyses.

4.2.3.5 Mission Analysis of Directing

The directing phase of the command and control cycle is generally a straightforward matter of distributing orders to subordinate units. The HFE mission analysis for the directing phase should be simple because it will normally require little analysis. Once the subordinate units receive their orders, the headquarters that issued the orders will go into its monitoring phase. From the HFE perspective, the success of the directing phase depends primarily on the capacity of the CCIS's communications facilities to get orders from higher command to lower commands. If these facilities are intact, the HFE characteristics of the directing phase will not likely affect the success of an operation. The HFE-related factors that affect the success of an operation are in either the planning or monitoring phases of the command and control cycle. If an operation goes badly, it will be because of a bad plan, or failure to recognize and quickly deal with problems while monitoring the operation, or both.

4.2.3.6 Mission Analysis of Monitoring

In the monitoring phase of an operation, combat units send reports on their progress in the operation back to headquarters. In headquarters, monitoring consists of collecting and collating combat units' reports, and building up and maintaining a picture of the progress of the operation. If the operation runs into a problem, the commander can issue orders to adjust the original operation plan to overcome the problem. If the operation progresses as intended, the commander need take no action.

The activities in monitoring an operation are fundamentally different from the activities in planning an operation because monitoring activities are driven by substantially unpredictable events occurring outside the headquarters. In contrast, planning activities are driven by a fixed predictable process controlled from within the headquarters. Because of the different character of monitoring activities, the mission analysis of monitoring requires analysis methods different from ones that can be used for analysing planning.

When a military operation runs smoothly with little opposition, monitoring activities in a military headquarters are little different from those in a corporate headquarters monitoring the progress of a sales campaign. However, the nature of combat creates many conditions that make it difficult to monitor operations accurately, and in turn, to respond correctly to events that might hamper the progress of the operation. Communications with combat units are often physically difficult to maintain. Busy subordinate combat commanders may not have time to keep up a steady flow of timely situation reports to headquarters, and operation plans can seldom completely anticipate opposing forces' actions, so operations are always subject to unexpected events. Because of these factors, monitoring an operation is typically a case of working with incomplete information that is, to varying degrees, out-of-date. Consequently, many command decisions are made with uncertainty about the correctness of the decisions.

There is no generic analysis of monitoring activities that is in any way equivalent to the generic analysis for planning. The headquarters activities in the monitoring phase of an operation depends very much on the types of forces that are under command, and how much opposition is being faced by these forces. The following describes possible approaches to analysing the monitoring phase of operations based on the types of forces under command.

4.2.3.6.1 Analysis Based on Weapon System Mission

If the force under command is made up of a single type or class of weapon system, the sequence of command and control monitoring activities can be based on the weapon system's mission profile. For example, if the force under command consists of bombers, command and control monitoring activities can be described by the command and control activities that occur in each phase of the bomber's mission profile. This approach can be extended to any combat element made up of similar weapons systems that operate in a few fixed mission profiles. Examples of systems suited to this approach include naval patrols, most combat aviation missions, and single land force combat arms missions such as missions by armoured, infantry or artillery forces.

Beevis, *et al.* [7], in their section titled Mission and Scenario Analysis, describe two mission analysis methods for describing a weapon system mission. One method, called the narrative mission description, consists of developing a narrative description the sequence of events in a mission. The second method, called the graphic mission profile, is a graph of some system-relevant variable plotted against the time course of the mission. The graph is annotated to show the occurrence of key events during the mission.

Beevis *et al.* use a close air support mission to give an example of this narrative mission description method. A mission narrative starts with descriptions of the main phases of the mission. Each phase is then broken down into descriptions of the events within each phase. The process of successively breaking down mission events into sub-events can be carried out to any level of detail one wants. However, keeping in mind the principle that a mission analysis should be to get a general overview, breaking the mission description down into perhaps two or three levels of detail should be sufficient.

A narrative mission description should provide a good basis for identifying command and control monitoring activities. Although a mission narrative will not likely explicitly identify command and control activities, subject matter experts can identify points where command and control activities occur during the mission, and in turn, describe the activities. Doing this will yield a separate narrative description of the command and control activities during a mission, and simultaneously link the command and control activities to events in the weapon system's mission.

The graphic mission profile is probably less useful than the narrative mission description for extracting a command and control mission analysis. The variable or variables being graphed against time may have little direct influence on command and control activities. For instance, the example of a graphic mission profile given by Beevis *et al.* graphs flight altitudes against time for a close support attack aviation mission. The graph itself would provide little basis for extracting command and control mission information. This leaves one with only the graph's annotations as a source of information. These are typically too short and cryptic to be of much use. So, the narrative mission profile will likely provide the best starting point for a command and control monitoring profile base on a weapons system mission profile.

4.2.3.6.2 Analysis by Representative Combat Scenarios

When the forces under command contain a mix of weapon systems, and especially when the forces are some combination of sea, land or air forces, a weapon system's mission profile will not provide a basis for describing the monitoring phase of command and control. Because different weapon systems have different mission profiles, there will be no common mission framework for describing the command and control monitoring activities. Now, monitoring activities have to be described by representative scenarios of the events in the command and control system's headquarters. For operations of any size, composed of mixed forces, the command and control headquarters will often be a hierarchical network of different headquarters.

The events in a headquarters during monitoring can be described from the messages between lower and higher headquarters. The contents of these messages drive nearly all headquarters monitoring activities. On arrival at a headquarters, each message triggers the following basic sequence of activities in the headquarters operations cell.

1 Log the information received in the message.

2 Decide what action to take in response to the message.

3 Perform the action decided upon from the message.

4 Formulate a message reflecting the action taken.

5 Log and send the message to another headquarters.

Almost everything that happens not only in headquarters, but also in the operation during the monitoring phase of an operation can be described from the contents of the repetitions of this sequence.

The sequence of actions in response to receiving a message is also fairly universal. The action sequence just described can be broken down a little further without losing generality by classifying the actions taken in response to different messages. Messages divide into two basic categories: 1) messages received from lower headquarters, and 2) messages received from higher headquarters. Further, the contents of messages from each level of headquarters can be classified into several basic categories.

Messages from lower headquarters break into the following categories.

1 *Information messages.* These break down into three types:

  1.1 battle situation information that needs to be immediately passed to higher headquarters: for example, reporting enemy contact;

  1.2 non urgent battle situation information that can be accumulated into a regular situation report that will be sent to higher headquarters at a scheduled time, and

  1.3 regular status reports (on such things as supplies) that will be used to update the headquarters' own status reports.

2 *Requests.* There are three types of requests:

  2.1 a request for information, which can be either an information request that can be answered from within the headquarters, or an information request that can only be answered by a higher headquarters,

  2.2 a request for combat support from another combat arm or logistics unit (air support or transport for example),

  2.3 a request for approval to take some action not covered in the current orders.

Messages from higher headquarters break into the following categories.

1 *Orders.* Orders trigger the planning process that eventually produces orders that will be passed down to lower headquarters.

2 *Response to a request.* A response to a request may be either a response to a request that originated in the headquarters, or a response to a request that originated in a lower headquarters.

3 *Information.* Information from higher headquarters is usually in the form of regular reports about the general operations situation. Occasionally, the information is about some matter that the higher headquarters thinks the lower headquarters should know about before one of the regular higher headquarters reports is due.

4 *Request for information.* A request for information from higher headquarters is usually about some matter that falls outside the scope of regular reports to headquarters, or about some matter that should have been covered in one of the headquarters' regular reports.

Most of the messages traffic to, from and through a headquarters is between the headquarters and lower headquarters. This is mainly because a headquarters normally has several headquarters below it, but only one higher headquarters to which it reports.

Figure 4 shows a flow chart illustrating the flow of message traffic through a headquarters. The chart shows two flows beginning with messages coming into the headquarters from either higher or lower headquarters. When a message comes in, its contents are logged, and in most headquarters, plotted on or referenced to a tactical map. Each message then goes through a sorting process to decide what action to take in response to the message. Whatever action is taken, it will inevitably generate another message that will be sent on to either a higher or lower headquarters.

The system of message communications through a headquarters provides a basic framework for describing a headquarters monitoring activities. In headquarters that control several types of combat elements, the messaging system may be the only possible medium for analysing the monitoring phase of operations. In cases where a weapon system mission profile can be used to describe the monitoring phase, the messaging system may provide an adjunct analysis.

A description of the system that a headquarters uses to handle messages does not include information about the dynamics of the communications through the headquarters. Unless an operation is thoroughly a 'set-piece battle', the sequence of events portrayed by the flow of messages through a headquarters will be a series of random message occurrences. The only way one can describe these occurrences systematically is by a narrative description of the operation scenario combined with the statistical parameters that represent the stochastic properties of the flow of messages. Getting a set of narrative descriptions representative of the monitoring process in a typical operation is usually fairly easy, but statistics on the stochastic properties of the monitoring process will seldom be available. Collecting the data needed for these statistics requires effort that should be left to later analyses (see HFE Function Analysis in Section 4.2.4, page 49).

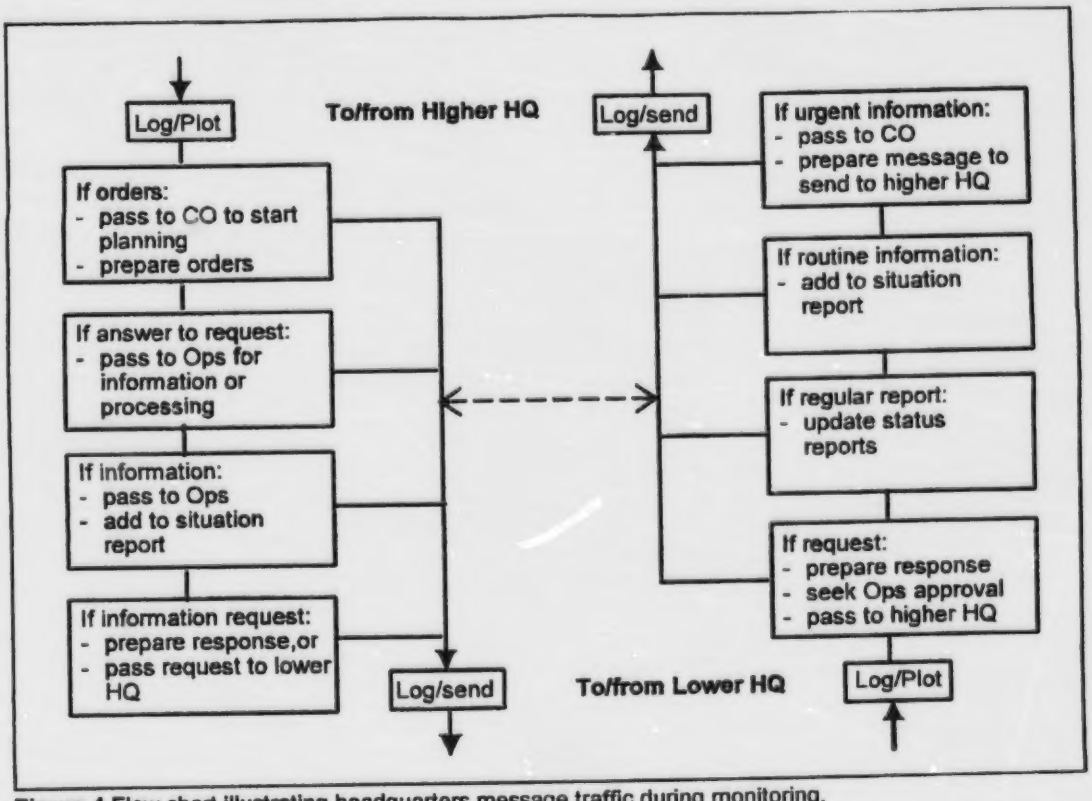

Figure 4 Flow chart illustrating headquarters message traffic during monitoring.

4.2.3.7 Mission Analysis of Multilevel Command and Control Systems

The mission analysis methods just described were formulated for a single headquarters. These approaches will be sufficient if the CCIS to be developed is to be essentially a within-headquarters or single-headquarters system. If the CCIS is to serve a multilevel hierarchy of headquarters, the mission analysis has to be extended to cover not only what happens within individual headquarters, but also what happens among different headquarters levels. The following describes some analysis approaches for dealing with this situation.

In the planning phase of the command and control cycle, the overall planning process can be done as separate analyses of the individual headquarters, with an additional analysis linking the planning process in each headquarters to the next. Except for one complication, the planning process in a headquarters proceeds independently of other headquarters' planning activities. The complication comes from what happens in actual practice compared with what a formal analysis might suggest. The complication can be seen from considering a hierarchy of headquarters, each with a planning process something like the one in Figure 2.

In theory the sequence of headquarters planning processes is a chain in which the highest headquarters completes its plan and then issues orders to its subordinate headquarters, each of which then does its plan and issues orders to its subordinate headquarters, and so on down the chain of command. This implies that each level of headquarters only starts planning after getting an operation order from the higher headquarters. If multilevel planning were done this way, the planning process for an operation would be much longer than would be tactically sound, and much longer than planning takes in practice.

To overcome the delays inherent in a lockstep planning process, multilevel command and control systems normally have systems of both formal and informal devices for shortening the planning process. The warning order is usually the main formal device for shortening planning. When a higher headquarters knows the outline of the next operation, it can issue a warning order outlining the operation to the next lower headquarters. The lower headquarters can then begin preparing a provisional plan that will be completed or finalized when the operation order is received. Having a system of warning orders permits each level of headquarters to get its planning under way well before higher levels of headquarters complete their plans.

The system of situation reports that regularly pass up and down a chain of headquarters provide another device for minimizing planning time. The accumulated contents of situation reports give a commander and staff a good basis for predicting the next operation, and therefore a basis for doing preliminary planning before even a warning order is received. Finally, command and control organizations generally maintain and encourage a system of personal liaison contacts that keep commanders and staff informed about the general operations situation. These liaison contacts provide additional sources of information for doing planning tasks before receiving the operations orders that formally start the headquarters planning process.

The mechanisms that allow multilevel command and control systems to minimize planning time may not be easy to document in a system analysis - mainly because the mechanisms themselves are not documented. Nevertheless, it is important to allow for them to ensure two things: 1) the mission profile of the planning phase gives a reasonably accurate estimate of how long it takes to complete the overall planning process, and 2) the mission profile includes the mechanisms that permit each headquarters to do advance planning.

Like planning activities within a headquarters, monitoring activities can be described within each headquarters independently, with communications to and from other headquarters being treated as system inputs and outputs. Considerations of interactions among headquarters should focus on the flow of information up through the command and control hierarchy. The factor that is by far the most critical to a commander's comprehension of an operation's progress is the timeliness, accuracy and completeness of information about the situation at the combat level. At a high level headquarters, this information is a distillation of information coming up through several levels of headquarters. As the information handling system in Figure 4 should suggest, by the time information from combat levels gets to a high level headquarters, opportunities for delays and inaccuracies have been many. Not having a clear picture of the operations situation is probably the thing that makes a commander most vulnerable to making bad tactical decisions. Therefore, analyses of command and control system features that control the timeliness and accuracy of

inter-headquarters information flows are an important part of the mission analysis of a multilevel command and control system.

A final possible conception of multilevel command and control that may be useful in mission analysis consists of treating the system as a set of nested control loops. Figure 5 illustrates this approach. The object of the control system is the combat element represented at the bottom of the diagram. The inputs to the combat element are controlled by the combat element's command element. In turn, the combat element's outputs feed back into the combat elements command function. Above the combat command function there is a hierarchy of control loops representing tactical command, strategic command, and on up to the organization's national headquarters. This system is a form of optimum control [26]. In it, combat command exercises immediate control over the combat element. Higher levels of command exercise increasingly broader and longer term control. The highest levels of control set goals, doctrine, and operating standards.

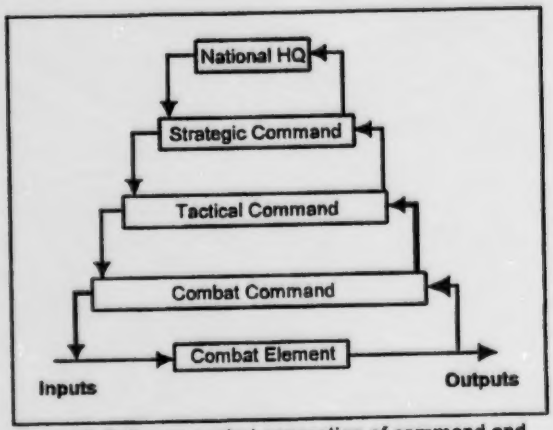

Figure 5 Optimum control conception of command and control.

A primary function of optimum control is to prevent its opposite condition: sub-optimization. In a multilevel command and control system, describing the nature of the control that each level of the control system exerts is an important part of the analysis. In a mission analysis, this can be done by adding a description to each box in Figure 5 that outlines the optimizing influences of each level in the hierarchy. The descriptions should be done with two purposes in mind. The first is to ensure that later, the CCIS design does not optimize command and control systems at one level of command at the expense of sub-optimizing systems at other levels of command. The second purpose is the related one of ensuring that the CCIS design preserves and respects the optimizing influences of the different levels of the command and control system.

Holley [27], in reviewing the post-World War II impacts of technology on command and control, points out two areas where exercise of command and control is vulnerable to what is being called here sub-optimizing. One area of vulnerability is in the amount and quality of the information that a CCIS makes available to a commander. With increasingly high speed wide band communications, it is increasingly possible to overwhelm high level commanders with large volumes of detail that (as Holley puts it) make little sense. The second area of vulnerability is the converse of the first. Having high speed wideband communications makes it possible for high level commanders (again as Holley puts it) to micro-manage the situation at the combat level. Both problems result from overzealous adoption of technology without respecting the long standing and well tried optimizing principles that govern command and control structures. One way to avoid making this mistake in a CCIS development is to make the analysis concept illustrated in Figure 5 a part of the system mission analysis.

4.2.3.8 Data Gathering Procedures for Mission Analysis

The mission analysis for the overall system engineering effort will normally be done by non-HFE personnel. In fact, it may be done by the acquisition authority as part of the project design phase. No matter who does the overall mission analysis, it should be possible to extract almost all the information needed for the HFE aspects of the mission analysis from the general system mission analysis. If the general mission analysis adequately covers information and factors expected to be important for later HFE work, a separate HFE mission analysis is unnecessary. If the general mission analysis has HFE gaps, these gaps can be filled in by gathering additional information from the user organization. Gathering this additional information can be done by using methods described in more detail in the section on task analysis (Section 4.2.5, page 73).

4.2.3.9 HFE Uses of Mission Analysis Information

Possibly the most important function of the mission analysis is to provide an overview of the command and control system. As detailed analysis and design work proceeds, even in a modest system, it will become harder to keep an overview of the system as a whole, and to keep the overall objectives of the system in mind. The mission analysis can provide a concise comprehensible description of the system, and consequently a framework for ensuring that all work stays directed at achieving the aims of the development project. Consequently, the depth of the mission analysis should be kept to that which will yield an understanding of the system as a whole.

The mission analysis should also provide the foundation for planning the subsequent HFE analyses: the function, task and operational sequence analyses. For the function analysis, the mission analysis will provide a skeleton structure identifying the command and control system's main functional components. Planning the subsequent function analysis can be done by filling in the details of the functions identified in the mission analysis. For the task analysis, the mission analysis will identify the groups of command and control personnel whose tasks will be analysed. Mission analysis will also provide an idea of the nature of the jobs different personnel do, and suggest which task analysis methods to use in the task analysis work. Mission analysis will also show if operational sequence analyses will be needed for describing the monitoring phase of the command and control cycle, and what types of sequence analyses would be most useful.

Finally, the mission analysis should provide a basis for identifying user performance measures for use in later design testing. Two basic measures of performance apply to command and control systems. One is the time needed to complete command and control tasks, particularly planning tasks. The other is the quality of command and control decision making. The time required to plan an operation is a traditional performance measure reflecting the fact that for two opposing forces, the one that has an advantage is the one that can 'get inside' the other's command and control cycle. Thus, any CCIS features that affect the time that command and control personnel need to plan operations are potential performance measures.

Quality of command and control decision making is an intuitively obvious measure of command and control performance. Although quality of command and control decision making is difficult to measure directly and objectively, several conditions are prerequisites to good decision making. For example, good command and control decision making requires timely and accurate information about the battle situation. The CCIS is the command and control system's medium for providing timely and accurate information about the battle situation. Therefore, any CCIS variable that affects information availability, accuracy, or timeliness is a potential system performance measure.

4.2.3.10 Summary of Mission Analysis

The following points summarize the contents of this section on mission analysis methods for command and control systems.

1 Mission analysis has two main purposes:

  1.1 to provide a comprehensible overview of what a system does and how it works, and

  1.2 to provide a foundation for planning subsequent HFE analyses, and for identifying design test performance measurements.

2 A mission analysis of a command and control system is best done by breaking the command and control cycle into three phases: planning, directing, and monitoring.

3 The mission description of the planning phase can be done by using 1) flow charts of the planning activities to show the nature and sequence of the activities, and 2) critical path charts showing the timings of activities, and the critical path of the planning process.

4 If the directing phase is limited to distributing orders to subordinate commands or combat elements, the directing phase should require no HFE analysis beyond identifying it as the transition into the monitoring phase.

5 There are three main approaches to doing a mission analysis of the monitoring phase.

  5.1 If the command and control system exercises control over a single type of weapon system, the monitoring phase can be described by identifying monitoring activities associated with different phases of the weapon system's mission profile.

  5.2 If mixed forces are controlled, monitoring activities can be described with representative mission scenarios derived from the flow of messages through the command and control system's headquarters.

  5.3 If the command and control system contains more than one level of command, the mission can be described with operations scenarios within individual headquarters, supplemented with an analysis of the hierarchy of command and control loops that govern relationships among levels of command.

6 Most or all of the information needed for the HFE mission analysis will be available from mission analysis data collected in other system engineering analyses, or from mission descriptions produced during project design.

7 The CCIS HFE Requirements (5.1.1.1) specifies the HFE mission analysis requirements. Documentation related to mission analysis is specified in the CCIS HFE Requirements DIDs titled Human Factors Engineering Analysis Report, Human Factors Engineering System Requirements and Function Allocation Report, and Human Factors Engineering Test Report.

8 The mission analysis results have three main HFE uses:

8.1 to provide a system overview;

8.2 provide a foundation for planning subsequent detailed HFE analyses, and

8.3 provide a foundation for defining design testing performance measurements.

The next sections describe the HFE analyses that follow from the mission analysis.

4.2.4 HFE Function Analysis

This section describes the uses of function analysis methods in CCIS development. Two types of function analysis methods will be described: 1) methods used in general system engineering, and 2) methods used specifically for HFE purposes. The section is divided into five main parts:

1 a description of the CCIS HFE Requirements specifications of function analysis;

2 a discussion of the purpose and nature of function analysis;

3 descriptions of function analysis methods applicable to CCIS development;

4 a discussion of using different function analysis methods for different phases of the command and control cycle, and

5 a description of data gathering methods for function analysis.

4.2.4.1 CCIS HFE Requirements Specifications for Function Analysis

Function analysis requirements are specified in CCIS HFE Requirements (5.1.1.2 and 5.1.1.3).

CCIS HFE Requirements (5.1.1.2) requires an analysis of the organizational structure of the command and control system. The analysis should include descriptions of structure of the command and control organization, the roles and duties of the command and control system's personnel, relationships among functional elements of the organization, and specifications of military doctrine that must be supported or preserved in the CCIS. The requirement for a description of the organizational structure can be met with the analyses that will be described in Section 4.2.4.3.6 Analysis of Functional Organization (page 68).

Depending on the circumstances of the project, a description of the doctrine that must be preserved in the new system could be obtained as part of function analysis work. It could also come from aggregating information from several sources such as the mission analysis, function analysis and task analysis activities. Wherever the information comes from, it is a good idea to create a single statement or document that specifies doctrinal constraints that the new system

must support and obey. During system design and testing, this statement will provide an important reference point for ensuring that the system conforms to doctrinal requirements.

The CCIS HFE Requirements (5.1.1.3) specifies function analyses that describe the command and control system's information flows, information management processes and decision making processes. The analyses are to include critical path and operational sequence data showing the timings and durations of activities in the command and control cycle, and identification of functions and sequences that are critical for achieving mission objectives. These requirements can be met with information from function flow or structured analyses like those that will be described in Section 4.2.4.3 General System Engineering Function Analysis Methods (page 54). Network analyses such as Petri net methods that will be described in Section 4.2.4.3.5 (page 66) would have limited value for meeting the function analysis requirement in CCIS HFE Requirements (5.1.1.3).

Although the CCIS HFE Requirements is not intended to constrain technical methods that might be applied to CCIS development, the constraints in CCIS HFE Requirements (5.1.1.3) are necessary to ensure that the HFE Plan is based on a thorough analysis of the functions that command and control personnel do to fulfill their duties. The types of function analyses specified in CCIS HFE Requirements (5.1.1.3) would meet this need. This does not mean that an argument cannot be made for using other techniques, if the other techniques meet the fundamental objective of understanding the roles and functions of command and control personnel.

The CCIS HFE Requirements (5.1.2) specifies that the results of the function and task analyses are to be further analysed to identify measures of user and command and control system performance and effectiveness. Methods for meeting this requirement will be described in Section 4.2.7 (page 121).

Documentation of the function analysis results is specified in the CCIS HFE Requirements DID titled Human Factors Engineering Analysis Report. Function analysis results are also needed for the DID titled Human Factors Engineering System Requirements and Function Allocation Report. In this report, the function analysis results provide the rationale for the function allocations and decision support requirements that are the main subject matter of the DID.

Function analysis results also provide part of the rationale underlying the HFE design approach described in the DID titled Human Factors Engineering Design Approach Document. The function analysis results also support the design tests described in the DIDs Human Factors Engineering Test Plan and Human Factors Engineering Test Report. Finally, the function analysis results will also contribute to the data item described in the DID titled Human Factors Engineering Design Review Report. Although the design review is aimed at comparing system design features with general HFE design standards, all HFE design standards presume that each design feature supports system functional requirements. Thus, the aim of HFE design review is to decide if the HFE design is optimum for meeting the feature's functional requirement.

4.2.4.2 Purpose and Nature of Function Analysis

In traditional HFE work, function analysis has played a minor analytic role whose purpose has never been well defined (for example, see Meister [5]). Typically, function analysis in HFE serves as a high level analysis of the system variables that influence or control what people do in a system. Often, function analysis is essentially a high-level analysis bridge going from mission analysis to detailed HFE analyses such as task analysis. In contrast, in other system engineering disciplines such as software engineering, function analysis has a much more important analysis role than it does in HFE. Related to this, the term function analysis in other system engineering disciplines may mean something different from what it means in HFE work. The interdisciplinary differences in the roles and usages connected with function analysis have considerable potential for creating confusion about the use of function analysis in the CCIS HFE plan. The following discussion is intended to clarify the relationship between function analysis usages in HFE and other system engineering activities, and to suggest how to deal with function analysis definitional problems in CCIS development. The discussion will cover the following issues:

1 the definition of function analysis

2 difference between the uses of function analysis in HFE and other system engineering activities

3 the role of function analysis in function allocation

4 the treatment of function analysis for the purposes of the CCIS HFE Requirements

4.2.4.2.1 Purpose and Definition of Function Analysis

The ultimate purpose of a general system engineering function analysis is to identify detailed system requirements. The analysis takes the system's overall performance objective, and breaks it down into separate performance components that can then be identified with human or machine elements capable of supporting the overall performance objective. Thus, in system engineering, function analysis is a primary method for developing detailed system requirements and subsequent design.

Although function analysis has no concise universal definition, it is essentially any method for analysing the behaviour of a system into its component and antecedent behaviours. The defining property of any function analysis method is that it describes a system's *behaviour* independent of the human and machine components that make up the system, or make up a system that is yet to be designed. Thus, Beevis *et al.* [18] define a function as a logical unit of the behaviour of a system, and define a function analysis as the successive breaking down of units of behaviour into their component subunits of behaviour. They further define function analysis as essentially a system engineering analysis rather than an HFE analysis.

4.2.4.2.2 Differences Between Function Analysis in HFE and in System Engineering

As briefly outlined in the last section, for (non HFE) system engineers, a function analysis is a system engineering design method based on analysing a system's behavioural objectives into component behaviours, ultimately for identifying ways to achieve the desired behaviours in a system design. At the same time however, the behavioural analysis in a system engineering function analysis is also the same basic behavioural analysis approach that has always been used to study human behaviour. That is, the analysis of human behaviour traditionally consists of breaking down units of behaviour into their smaller constituent units of behaviour. In HFE work, a function analysis often, perhaps usually, means a high level analysis of the behaviour of humans in a system or organization. And, as much as partitioning behaviour into constituent parts is a function analysis, HFE task analysis (see Section 4.2.5) is also a form of function analysis [28].

Comparing how function analysis is used in system engineering with how function analysis is used in HFE shows two major inter-discipline differences.

1 In system engineering, function analysis is primarily a design tool, while in HFE, function analysis is primarily an analysis tool.

2 In system engineering, the object of the function analysis is to understand the system independent of the system's components (human or machine), while in HFE, the object of the function analysis is to understand the behaviour of the human components of the system.

Because of these differences in function analysis aims and scope, HFE specialists need to take care to specify exactly what they mean when they use the term function analysis in connection with HFE work. HFE specialists equally need to take care to understand what other system engineering specialists mean by the term function analysis.

The broader scope of function analysis in system engineering compared with the scope of function analysis in HFE represents another major inter-discipline difference in function analysis usage. Because system engineering function analysis treats the behaviour of the system as a whole independent of the system's substance, it is possible to use any of many mathematical models to represent a function analysis - models with which HFE specialists may have little experience. In HFE, there are a fairly restricted variety of models used to represent function analysis methods. The reason for this comes largely from the fact that any HFE function analysis model of human behaviour will have implicit constraints imposed by the characteristics of human behaviour. In contrast, a system engineering function analysis is comparatively free of such constraints. In fact, a primary aim of function analysis in system engineering is to free the designer from constraints imposed by specific machine or human limitations. This gives the designer the widest possible scope for identifying the component behaviours that will be needed to achieve the system's objective.

Having identified the component behaviours, the system designer can go on to identify the human and machine elements needed to achieve the behaviour goals. Doing this is the role of function allocation.

4.2.4.2.3 Function Analysis and Function Allocation

Once the required component behaviours needed to make the system work are known, the designer can then look for the human or machine elements needed to produce the component behaviours. Looking for the human and machine elements that will be needed to support the required system behaviour is called function allocation. The role of HFE in function allocation normally consists of helping the system designer choose between human and machine for performing different system functions, but usually only for functions where the designer sees a human as a candidate for performing the functions. The description of HFE in function allocation will be left for the later section on HFE in System Design (see Section 4.3.1.3, page 130).

4.2.4.2.4 Function Analysis Usage for CCIS Development

The range of meanings and uses attached to function analytic methods across system engineering disciplines has considerable potential for creating confusion in managing both the HFE work and the system development work as a whole. To minimize this potential confusion for the purposes of this CCIS HFE Design Guidelines document, function analysis will be treated primarily as a system analysis method, normally conducted by non-HFE system engineering personnel, and used only in part to influence HFE design considerations such as determining which system functions to allocate to humans. This is the approach recommended by Beevis *et al.* [18].

Any specifically HFE function analysis work will be treated as a supplement to the main system engineering function analysis. This means that HFE function analyses should be conducted only when the main system engineering function analysis results leave gaps in the information needed for HFE work. Normally, this will mean using HFE function analysis methods for high level analyses of human activities in a system. Such analyses will usually consist of elaborating or extending the flow chart, critical path, and like methods described in connection with mission analysis in Section 4.2.3. For detailed analysis of human behaviour in a system, task analysis is usually the predominant HFE analysis method (see Section 4.2.5), with function analysis serving as a framework, or as a bridge from mission analysis to task analysis.

There are several reasons for giving the system engineering function analysis precedence over HFE function analysis. First, the precedence reflects the reality of the system engineering function analysis's importance in the overall system design effort. Giving precedence to the system engineering function analysis should also force the HFE specialist to study and understand the system engineering function analysis results, and by that, let the HFE specialist influence the function allocation effort later on. If the HFE specialist does not fully understand the function analysis results, HFE influence over function allocations will be limited to passively advising yes or no to isolated functions that the system designers propose to allocate to humans.

While this CCIS HFE Design Guidelines gives precedence to the system engineering function analysis, there is also no restriction on the use of any traditional HFE function analysis methods where they are needed. There are two reasons for not limiting the use of HFE function analysis methods. First, the system function analysis method chosen for the project may not be well suited

to account for properties of system behaviour that are peculiar to human behaviour. Some system function analysis methods are essentially abstract mathematical models that either neglect fundamental properties of human behaviour, or become too cumbersome when they try to deal with the complexities of human behaviour. If the chosen system engineering function analysis model cannot represent important aspects of human behaviour, the model will have to be supplemented with specific HFE analyses.

Another reason, possibly the most important reason for not restricting the use of HFE function analysis techniques concerns the nature of CCIS development. If one takes a strict view, a CCIS is a hardware and software system that has no human elements. That is, a CCIS is simply a communications network, usually containing a variety of computer elements. And though a CCIS's entire function is to support the human functions that make up the command and control system, it might be decided somewhere along the development line to take this strict view of a CCIS. This could conceivably lead to deciding that there is no need to include the human command and control functions in the CCIS system engineering function analysis. This decision might arise either as a conscious decision, or from not clearly thinking out what the scope of the development should be: that is, should the scope encompass the command and control system as a whole, or should it be restricted to the CCIS hardware and software system?

The CCIS HFE Requirements implicitly assumes that the aim of a CCIS development project is to improve command and control performance. However, this aim will not be met if the CCIS function analysis takes the existing command and control system for granted, and merely looks for better ways to move and store the data used by commanders and staff. For example, in any command and control system it is in principle a straightforward matter to identify the command and control system's data and data movements, and then construct a function analysis model to represent the data handling system. While this approach may yield faster and more reliable ways of handling existing command and control data, it is unlikely to identify ways to improve the performance of command and control system personnel. In fact, leaving commander and staff functions out of the analysis may well lead to a CCIS that leaves the command and control system no better off than it was before - perhaps worse off than before (see the earlier discussion on this point in Section 4.2.3.7).

With these considerations in mind, the following describes function analysis methods that are applicable to CCIS development, divided into two main groups: 1) general system engineering function analysis methods, and 2) HFE function analysis methods that may be used to supplement the general system engineering methods. In all cases, the analyses will be treated as methods for identifying system requirements or developing a detailed system design.

4.2.4.3 General System Engineering Function Analysis Methods

This section briefly describes system engineering function analysis methods that are applicable to CCIS development. The methods are ones that might be used in system analysis activities connected with identifying detailed system requirements. All the methods have the same ultimate objective: to provide a basis for identifying the combination of system components that will best

achieve the system's operational requirements. The methods proceed by decomposing the overall system (behaviour) requirement into more detailed component requirements until specific detailed requirements can be associated with corresponding human, hardware, or software elements that could potentially support each detailed requirement.

The description of each method concentrates on the method's applicability to CCIS development, and on how the method's results might be used for the purposes of HFE analysis. The following methods will be described.

1 Function flow analysis,

2 Structured Analysis and Design Technique (SADT™),

3 GRAI Method,

4 Petri Net, and

5 analysis of functional organization.

4.2.4.3.1 Function Flow Analysis

This is the oldest, and probably simplest method of analysing the sequence or sequence and flow of events during a system's operation. The results of a function flow analysis are represented in a function flow diagram. A traditional flow chart is an example of a function flow diagram. The diagram shows the sequence of events in a process, branches to alternative processes, and the decision rules that govern the path that the system takes through alternative different processes.

Figure 2 (page 37) and Figure 4 (page 44) in the previous Mission Analysis section (Section 4.2.3) show examples of high level function flow diagrams. Figure 2 shows the basic flow of activities in a command and control planning process. Figure 4 shows the basic flow of events in processing messages during the monitoring phase of the command and control cycle. In both cases, the function flows could be decomposed into more detailed charts by taking each box representing a function in the chart and describing the subfunction flows within the box. In fact, the planning process in Figure 2 could be a decomposition of the box in Figure 4 that represents receiving a message containing orders from higher headquarters.

Figure 6 shows another example of decomposition based on the message processing system in Figure 4. The example, a simple one, shows the decomposition of the process of handling an enemy contact report from a lower headquarters. The example illustrates two features of function flow analysis methods: 1) the propositional logic underlying the flow of events, and 2) a function numbering system for keeping track of the levels of decomposition.

To keep the decomposition manageable and comprehensible requires a hierarchical numbering system to identify levels of decomposition and to link sets of sub-functions to their superordinate functions. Figure 6 illustrates a common numbering system for identifying the hierarchical locations of different functions. The use of boxes containing brief function descriptions, the

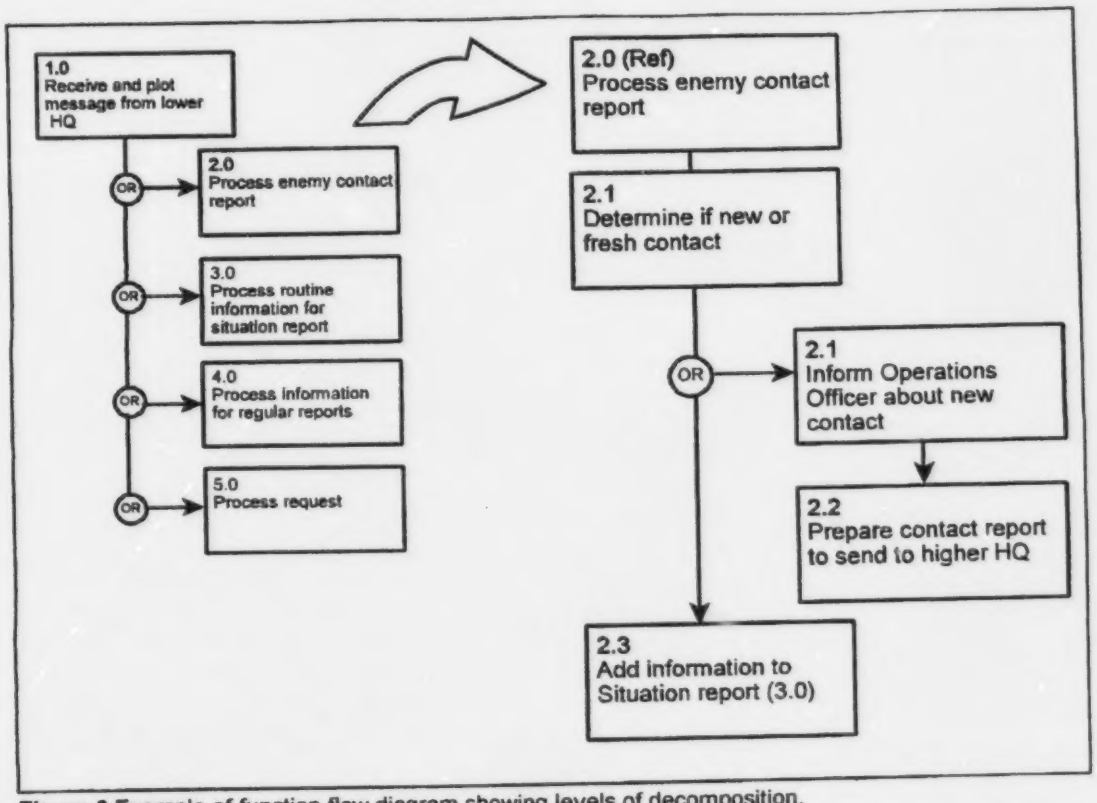

Figure 6 Example of function flow diagram showing levels of decomposition.

directed lines linking boxes, and the 'OR' gates for decision points are representations that are convenient for computer-based recording of function analysis data.

The traditional computer programming flow chart is another representation that is essentially a function flow technique. Beevis *et al.* [18] describe variations of this technique under a separate heading, *Information Flow and Processing Analysis*. Figure 7 shows flow chart example again using the headquarters message handling process. The flow chart analysis has the merit of being a familiar technique both to the analyst, and to many people, system users, for example, who have to read the output of the analysis. It is also an easy medium to use for collecting and recording the analysis data. For example, the original of the chart in Figure 7 was a hand-drawn chart on a roll of newsprint, produced by an analyst and a subject matter expert as they worked out and recorded the logical flow of message handling in a headquarters.

A function flow diagram can be used to represent any logical sequence that can be understood as a flow. For example, the functional flows can be human activities like the planning activities portrayed Figure 2, or just the information flows in the planning process, or the flow of a certain type of paperwork in a planning process. As these examples suggest, the same system can be

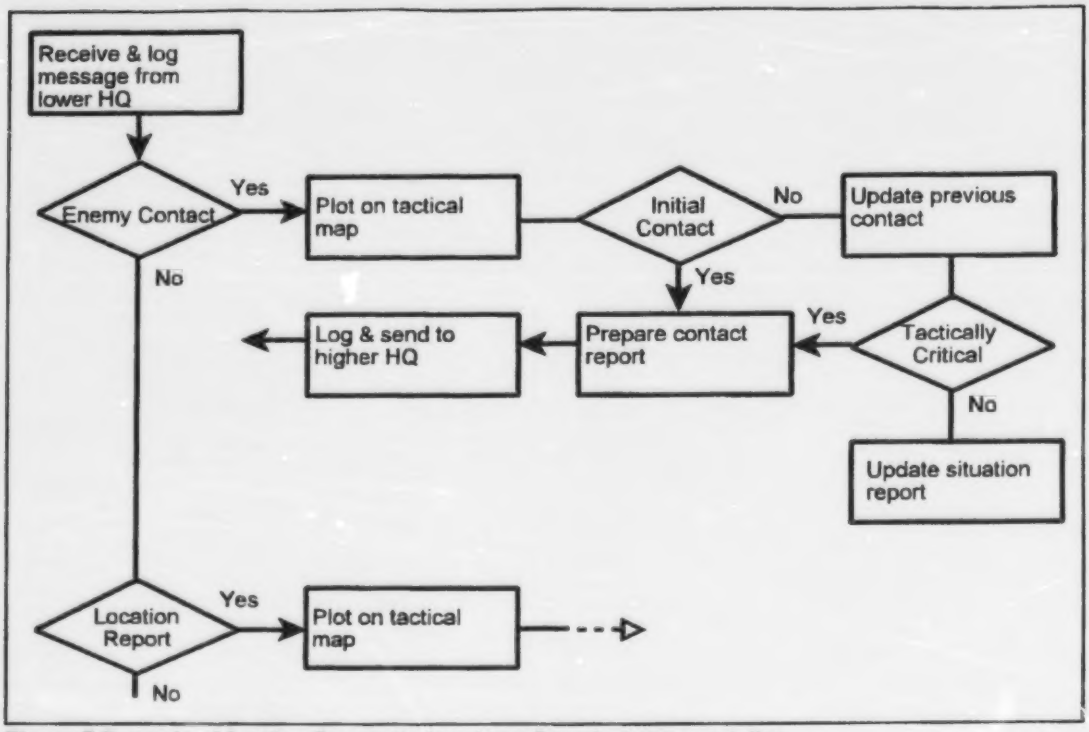

Figure 7 Example of function flow analysis using a flow chart representation.

represented by several different function flow diagrams, each representing a parallel decomposition of functions - which might be thought of as orthogonal to the hierarchical decompositions. Figure 8 illustrates this idea.

Figure 8 shows three notional flow diagrams each representing an aspect of the same underlying process. The leftmost one shows activities. The next shows data processes. The rightmost shows paperwork. The three diagrams are congruent, but with some elements in the data processes and paperwork diagrams shown with dashed lines. The elements shown with dashed lines represent functions in the activities diagram that have no corresponding functions in the data processes or paperwork flows.

The idea of parallel function flows is important to keep in mind for applying function analysis results to HFE work. First, the existence of parallel function flows is a necessity coming from the fact that a function flow can only represent the flow of one thing. The flow can be of human activities, or of information, but it cannot be sometimes here activities and sometimes there information. It follows from this that a single function flow representation is very likely an incomplete representation of how a system works. A single function flow representation is a representation of how one aspect of a system works. This is an important point for doing HFE work in a CCIS development.

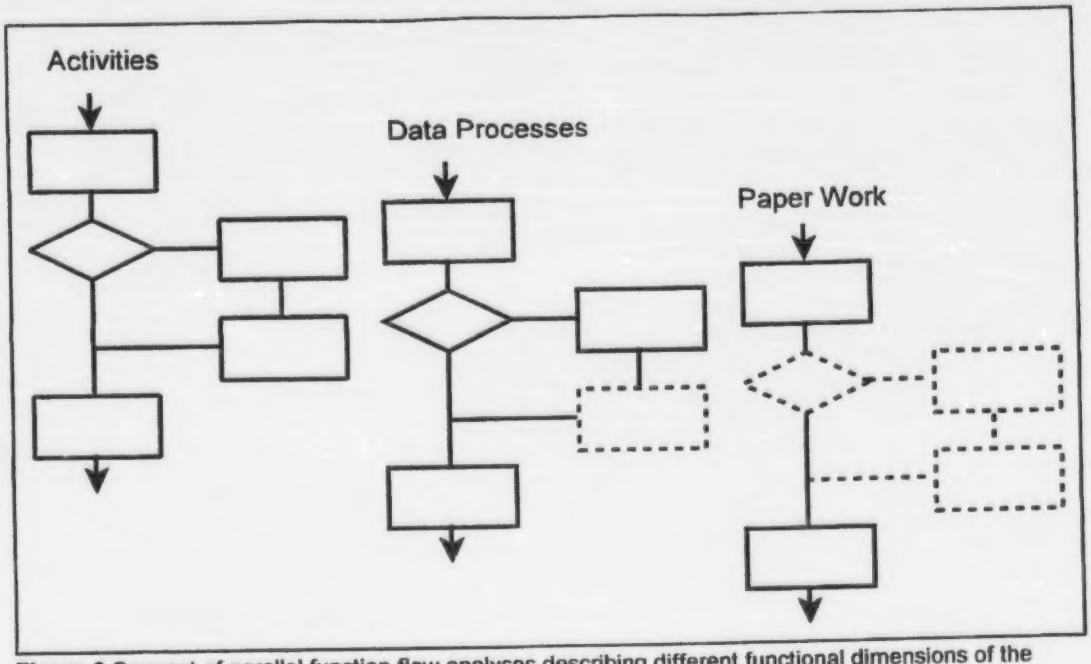

Figure 8 Concept of parallel function flow analyses describing different functional dimensions of the same system.

In a CCIS development, the obvious system engineering function flow analysis to do is an analysis of data flows. Although such an analysis would start with an analysis looking much like the activities flows in the planning process representation in Figure 2, the process of decomposition and abstraction may soon lead to information flow diagrams that tell very little about how command and control personnel actually use the information represented in the diagrams. To understand what personnel do requires either a parallel analysis describing what personnel do with the information flows, or another device that connects personnel performance with the information flows. Whatever device is used, the HFE specialist must ensure that whatever comes out at the end of the overall function flow analysis, it includes a component analysis of personnel functions. Without this human function analysis, the HFE specialist will have little basis for doing subsequent HFE tasks such as contributing to function allocation decisions.

A function flow analysis can be done either on an existing predecessor system or on a system yet to be built. In the latter case, the analysis takes the gross system function and breaks it down into the functions that would logically be needed to realize the overall system function. In CCIS development, the development objective is usually to improve an existing command and control system. Therefore, the analysis usually starts with an analysis of the predecessor system, and then aims at finding improved ways to support different functions. In this context, it is important for the HFE specialist to ensure that the analysis of command and control activities identifies those functions in the predecessor system that must be preserved on the successor system.

4.2.4.3.2 Sequence and Timing (SAT) Diagrams

A sequence and timing diagram is a type of function flow diagram. It shows the sequence events in the performance of a single system function, in particular, the sequence in which a system's subsystems are begun when the system performs one function. Although the method is commonly used to analyse administrative paperwork systems, it is not often described in the literature (Beevis *et al.* [18] being an exception). However, it will be briefly described here because it can be useful in some command and control environments. Its usefulness lies in its ability to isolate a specific function, and trace the events that occur during execution of the function. Figure 9 illustrates this using the headquarters message flow example.

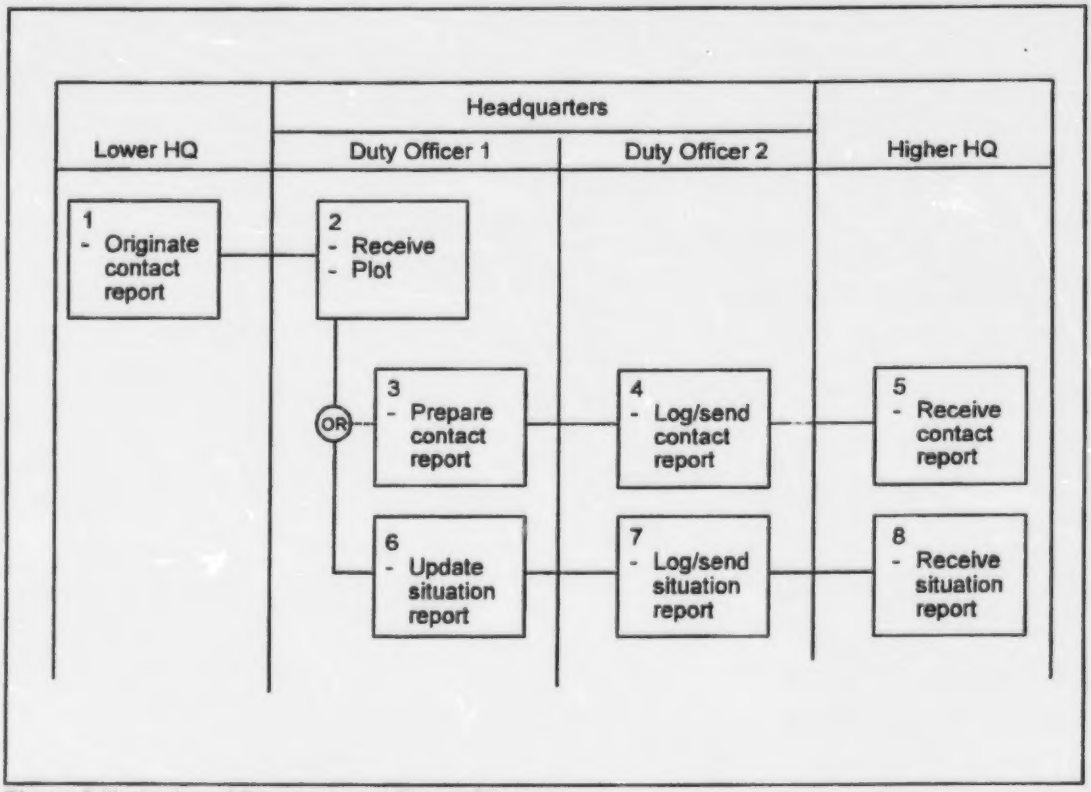

Figure 9 Illustration of Sequence and Timing Diagram.

Figure 9 shows a sequence of events in processing a contact report, similar to the message handling flow diagrams in Figure 4 and Figure 7. Figure 9 is different from the earlier two examples by showing two officers handling the message traffic instead of just the one implied in the earlier examples.

Although Figure 9 shows essentially the same events as the earlier flow process diagrams, the events are represented from a different perspective. In the SAT diagram in Figure 9, only one

function is portrayed: that of handling a contact report. In addition, the SAT diagram partials out the subsystems involved in processing a contact report. The diagram shows four subsystems: the lower headquarters, the two duty officers in the middle headquarters, and the higher headquarters. These subsystems are represented as columns in the diagram. Although the subsystems in this example are all humans, subsystems in SAT diagrams can be either human or machine elements. In Figure 9, the subsystems are represented by the column headings, and the contents of the columns show the actions performed by different subsystems. The sequence of the actions is shown by the relative positions of the actions in the columns.

The primary usefulness of an SAT diagram is to show a trace of the sequence of events in a critical process, and to show how the functions making up the process are shared or divided among different subsystems. These virtues are also the SAT diagram's limitations. By showing one process, the SAT diagram does not show all the other things that may be going on in the system simultaneously. To remain comprehensible, the SAT diagram is restricted to simple sequential processes, ones that do not contain complex branching or looping.

Because an SAT diagram is an after-the-fact representation of how subsystems interact to perform a function, it does not serve the primary purpose of a function analysis method. That is, it does not break down overall system function into sub-functions until the sub-functions can be allocated to subsystems. An SAT diagram can only be used when the subsystems are already defined. SAT diagrams are therefore limited to analysing existing systems, or analysing the feasibility of proposed systems. Overall, an SAT diagram is best suited to situations where a particular function in a real or proposed system has to be analysed in isolation from other functions.

4.2.4.3.3 SADT™ and IDEF

Structured Analysis and Design Technique (SADT™) and IDEF are two closely related structured analysis methods. The following describes each method in turn.

4.2.4.3.3.1 Structured Analysis and Design Technique (SADT™)

Structured Analysis and Design Technique (Ross and Schoman [29]; Marca and McGrowan [30]) is designed to provide a system development team with a framework for defining system requirements. In part, SADT™ is a graphical model for producing a functional description of a system, and in part a framework for communications and cooperation among members of a team of analysts. SADT™ is specifically designed to support the analysis of large complex systems.

SADT™ conceptualizes a system function as an entity that has a set of *outputs*[12] controlled by three sets of inputs. The system entity has three inputs: 1) *inputs* of information and materials, 2) *controls* consisting of goal definition actions or objects, and 3) *mechanisms*, the physical

[12] The italicising of terms indicates that the terms are standard SADT™ terms.

devices and human actions that perform the function. Figure 10 illustrates how a function with its inputs and outputs are represented in a basic SADT™ diagram.

In Figure 10, the box represents the function being analysed. The sides of the boxes represent the function's *interfaces* with its inputs and outputs. The positions of interfaces are standard; that is, *inputs* on the left side, *controls* at the top, *mechanisms* at the bottom, and *outputs* on the right.

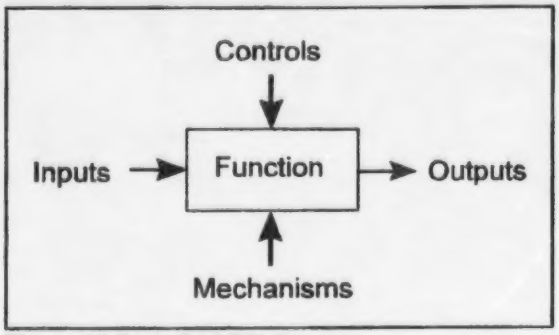

Figure 10 Basic SADT™ diagram.

Figure 11 gives a command and control system example of the types of things that would be included as part of each interface. The function portrayed in Figure 11 is the overall command and control function of a headquarters. The inputs to the command and control function are data on the battle situation, data on the status of supplies, and generally any other information that the commander and staff use to make command and control decisions. The controls include orders from higher headquarters, along with standing directives such as standard operating procedures and tactical doctrine. The outputs are orders to lower headquarters, and data as messages and reports sent to both higher and lower headquarters. The command and control mechanisms in Figure 11 are the commander and staff, and the CCIS.

In Figure 11, the orders received by the headquarters are treated as controls. These orders might equally well be treated as input data. They are treated as controls in Figure 11 on the grounds that receipt of orders generally starts a command and control operation in motion. The orders then treated as a set of standing orders until the operation is completed.

In an SADT™ analysis of a command and control system, Figure 11 would form the starting point of the analysis. Subsequent analysis consists of decomposing the base diagram in Figure 11 into component diagrams describing the sub-functions needed to support the system's overall function. Figure 12 illustrates the analysis with a decomposition of an army headquarters function

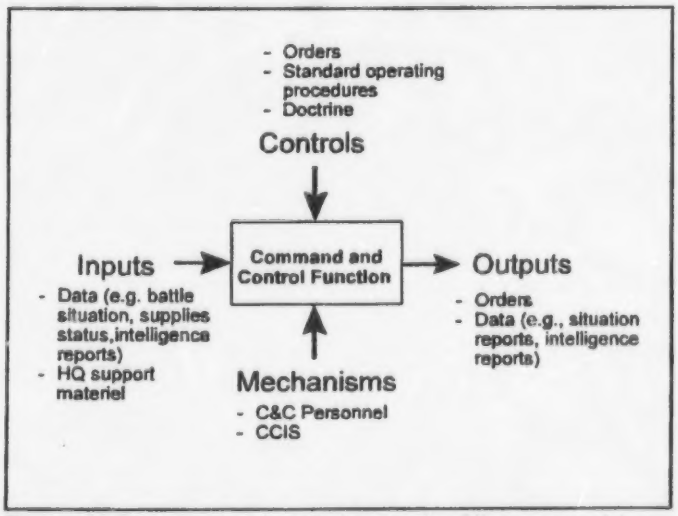

Figure 11 Command and control example of items comprising SADT™ inputs, controls, mechanisms and outputs.

broken down into a command function, and supporting G1 Personnel, G2 Intelligence, G3 Operations, and G4 Supply functions[13]. The breakdown in Figure 12 represents a first level of decomposition. The analysis would continue by breaking down each of the function boxes in the first level into their respective sets of sub-functions, and so on until there is no more to be gained by further decomposition.

In a completed SADT™, all the function boxes are linked through their input and output interfaces. These links are suggested in Figure 12 by lines going in the left side and out the right side of each box. The controls and mechanisms supporting each function are not shown, but implicitly, every function would have its own set of controls and mechanisms.

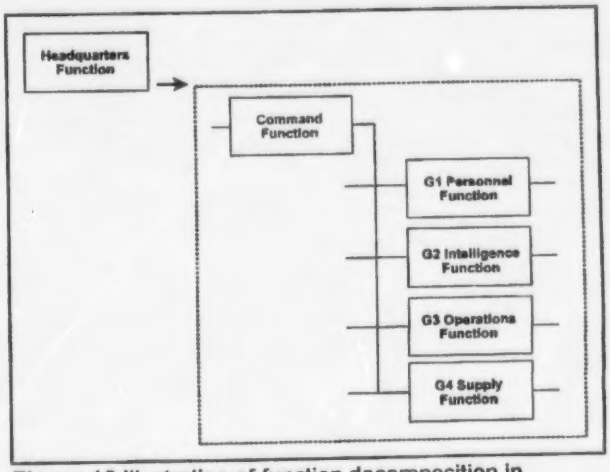

Figure 12 Illustration of function decomposition in SADT™ .

The analysis of an existing system would include identifying the current mechanisms supporting each function. However, the ultimate system development goal of the analysis would be to provide the basis for identifying improved mechanisms for supporting each function. Thus, the output of an SADT™ analysis defines the requirements for the mechanisms needed to support the different functions identified in the analysis. Identifying the specific mechanisms would be done later in a function allocation exercise.

The SADT™ package includes sets of rules defining such things as functions, inputs, outputs, controls and mechanisms. It also contains a collection of methods for conducting the analysis, verifying the results, and organizing the results for convenient reference. The package is clearly intended for use by teams of analysts who are analysing large systems.

SADT™ is described in Beevis *et al.* [18] with an orientation towards applications to military development projects. They also give references to several military examples of SADT™ applications. In addition, Beevis *et al.* describe SADT™'s limitations in HFE applications. Two important limitations are the following.

1 The SADT™ analysis does not show sequence and timing relationships among functions. Other analyses such as function flow and time line analyses are needed to describe the sequence and timing properties of the system.

2 The results of an SADT™ are not independent of the backgrounds and perspectives of the analysts.

---

[13] Although the nomenclature varies among military organizations, all command and control organizations include these basic personnel, intelligence, operations and supply functions.

The first limitation is implicit in the nature of the analysis. The second limitation is probably a limitation of any function analysis method. One way or another, the analyst has to apply his or her individual general technical knowledge and judgement to deciding such things as how a function should be divided into sub-functions, or when various function inputs, controls and the like have been sufficiently elaborated. SADT™ contains a formal procedure for independent reviews of analysis results. These procedures moderate the effects of individual biases, but they do not eliminate whatever collective biases the analyst team might have.

4.2.4.3.3.2 IDEF

IDEF is the military equivalent of SADT™, with the addition of two components (Ross [31], Luczak [32]). The components of IDEF are denoted as IDEF 0, IDEF 1, and IDEF 2; where IDEF 0 is equivalent to SADT™. IDEF 1 is a data model formulation parallel with the function model formulation in IDEF 0 (and SADT™). IDEF 2 provides a dynamic representation that models the execution of system functions. However, since there appears to be little literature on its use in command and control function analysis problems, IDEF 2 will not be discussed any further.

Figure 13 illustrates the data model conception embodied in IDEF 1. The structure shown in Figure 13 is similar to the function model used in IDEF 0 and SADT (see Figure 10). However, as shown in Figure 13, the box represents data, and the inputs and outputs are functions rather than data or other objects. Controls in IDEF 1 are also different from controls in IDEF 0. In IDEF 1, controls represent the system data model that governs how data are produced and used in the system. Mechanisms in IDEF 1 have the same role in principle as mechanisms in IDEF 0, except that they are mechanisms for dealing with data rather than mechanisms for supporting functions.

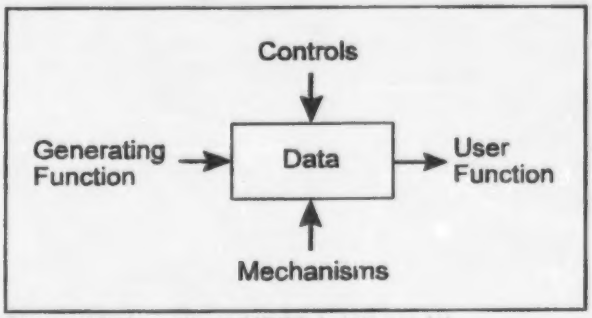

Figure 13 Illustration of IDEF 1 data model.

The limitations of IDEF 0 are the same limitations just described for SADT™. These limitations presumably also apply to IDEF 1.

4.2.4.3.4 GRAI Method

The GRAI method is intended for the analysis of decision making and information handling activities that go into planning and controlling manufacturing production systems. Although intended for manufacturing applications, the GRAI method has features that could make it useful

in command and control applications. The GRAI method is briefly described in Luczak [32], where references are also given to original sources.

The GRAI method divides a manufacturing organization into three components: 1) the decision system, 2) the information system, and 3) the physical production system. In addition, the GRAI method defines two types of functional activities: 1) decision activities, and 2) execution activities. Two logic models, one for decision making and one for execution, are defined for describing the logical operations underlying particular instances of each type of activity. These models are called GRAI networks.

A GRAI function analysis consists of producing a quasi formal description of how different organization functions use the information and decision making systems to cooperate during the production planning period leading up to realtime production activities. This GRAI analysis description is represented by a grid whose rows correspond to different amounts of planning lead time, and whose columns correspond to different organization functions. The cells in the grid contain descriptions of the tasks that different functions perform. Interactions among cells are represented by notational arrows that correspond to decision making and information relationships. The task described in any one cell can be described in detail by using one of the GRAI network models: that is, the decision making network if the task is a decision making task, or the execution model if the task is an execution task. Figure 14 gives a notional illustration of a GRAI grid.

The illustration of a GRAI method grid is based on extracting some of the cells in a more complete illustration in Figure 12.34 in Luzak [32]. The illustration in Figure 14 is not a complete representation of Luczak's example. It is only complete enough to demonstrate the idea of a GRAI grid representation. As explained earlier, the rows of the grid represent time horizons leading up to realtime production activity. The columns represent organizational functions, in this case, Sales, Planning, and Buying. The cells in Figure 14 briefly describe the tasks that different functions perform at different time horizons.

The solid and dashed arrows in Figure 14 are the notations used to show decision making and information sharing relationships. The solid arrows represent transmission of decision outcomes. A solid arrow is essentially a directive, which can be either a direction to do a task, or a direction implied by the organization's standard operating procedures. In Figure 14, the solid arrows show how decision making tasks cause or influence the performance of other tasks.

The dashed arrows in Figure 14 represent

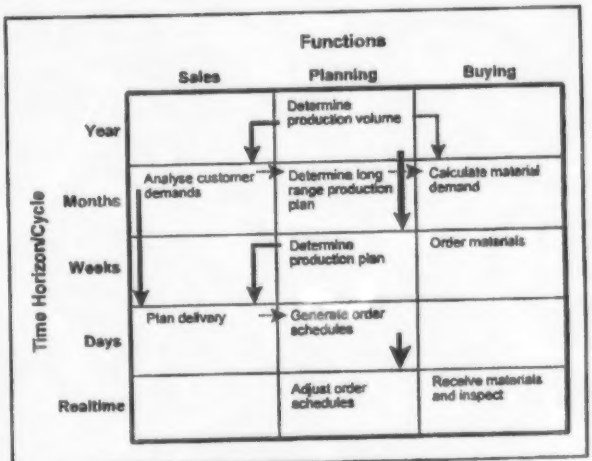

Figure 14 Illustration of GRAI method grid (after Figure 12.34 in Luzak [32]).

information flows. Information flows can have different roles that are not differentiated in the notation. Some information flows represent information that a function needs to do a task. Other information flows represent information that triggers a task or causes an adjustment in the task activity.

As the wordings describing the tasks in Figure 14 suggest, tasks in a GRAI grid are described at a fairly gross level of detail. In a GRAI analysis, tasks can be analysed in more detail using one of the GRAI networks. These networks will not be illustrated here. Illustrations can be found in [32]. Since the GRAI networks are particular forms of structural function analysis techniques, there is no reason that other function analysis methods could not be used to analyse GRAI grid tasks in detail.

According to Luczak, the GRAI method can provide a comprehensive top-down analysis of the organization-wide decision making processes that lead to production activities. A major virtue of the GRAI method is its ability to link all organizational decision making activities down to the assignment of production tasks to individual work systems or individual personnel. The structure and content of the analysis make it useful for identifying information system requirements and related software requirements.

The GRAI method is probably most useful for analysing the macro functions underlying an existing organization's operations management system. Although theoretically capable, the GRAI method is probably not practical for doing the types of detailed analyses that methods like function flow diagrams, SADT™, and the like can handle. On the other hand, these other detailed analysis methods are not well suited for doing three things that are fundamental to a GRAI analysis, namely: 1) accounting for the characteristics of high level organizational decision making, 2) describing how decision making and information flows at different levels of the organization interact to influence realtime operations, and 3) describing organizational planning and decision making by time horizons. Other analysis methods virtually ignore these aspects of operations management.

Although the GRAI method comes from analysing production systems, imagining that it could translate to a command and control environment is easy to do. For purposes of understanding the processes, the planning, directing and monitoring activities in a command and control operation are not very different from the planning and decision making activities in a manufacturing operation. As suggested by the GRAI method advantages described in the previous paragraph, a GRAI method analysis of a command and control system would permit analysis of three important aspects of command and control that are typically left out of other types of analyses. These three aspects are:

1 the nature of the problem solving and decision making that make up the early stages of developing a tactical operations plan;

2 how command and control decision making and information flows at different levels of the command and control organization interact in planning tactical operations, and

3 how the different planning time horizons of different levels of command and control affect decision making and planning activities at each level.

Although there appears to be no literature on applications of the GRAI method to command and control system analysis, this should be no barrier to at least looking at a particular command and control organization from a GRAI perspective. If nothing else, taking such a perspective would identify the gaps in information provided by whatever other function analysis methods are being used to analyse the particular command and control system.

4.2.4.3.5 Petri Net

A Petri net is a mathematical model that can represent a system's information processing properties. Figure 15 shows a very simple Petri net containing the basic elements needed to construct a Petri net.

Figure 15 shows the four elements needed to construct a Petri net:

1 *place*, denoted by a circle, represents a state or condition;

2 *token*, denoted by a small solid circle or dot in a *place*, indicates the state of a *place*: whether the *place* is on or off, for example;

3 *transition*, denoted by a bar, represents the occurrence of an event, and

4 *directed arcs*, denoted by an arrow; represent either the action of a *place* on a *transition*, or the action of a *transition* on a *place*.

Places can be identified as one or the other of two types: 1) *input places*, which are places whose directed arcs feed into transitions, and 2) *output places*, which are places that receive directed arcs from transitions.

A Petri net can represent two components of a system's information processing properties:

1 the state of the system at a point in time, represented by the distribution of tokens among places, and

2 how the state of the system will change when an event occurs at one of the net's transitions.

Changes in system states are governed by two fundamental rules that determine the triggering of transitions and the placements of tokens. These two rules are:

1 A transition will be triggered when all the input places from which it receives directed arcs contain tokens.

2 An output place will receive a token when any of the transitions from which it receives directed arcs are triggered.

Besides these fundamental rules, the Petri net modeller can define additional rules to represent particular characteristics of the system being modelled. For example, the modeller can impose durations and time delays on transitions, allow tokens to decay, impose probability conditions on transition events, and so on. In addition, the modeller can set up various placements of tokens, and then trigger a state change by placing tokens in selected places.

To apply a Petri net to a specific system analysis problem, the modeller has to translate system elements such as communications devices, machines, and human operators into Petri net places, transitions, and directed arcs. The modeller then has to define rules for the placement and movement of tokens in a way that will allow the net to represent the system's behaviour accurately. Once this is done, the modeller has to design the experiments needed to see how the model will behave under different system operating conditions, or under different system design options.

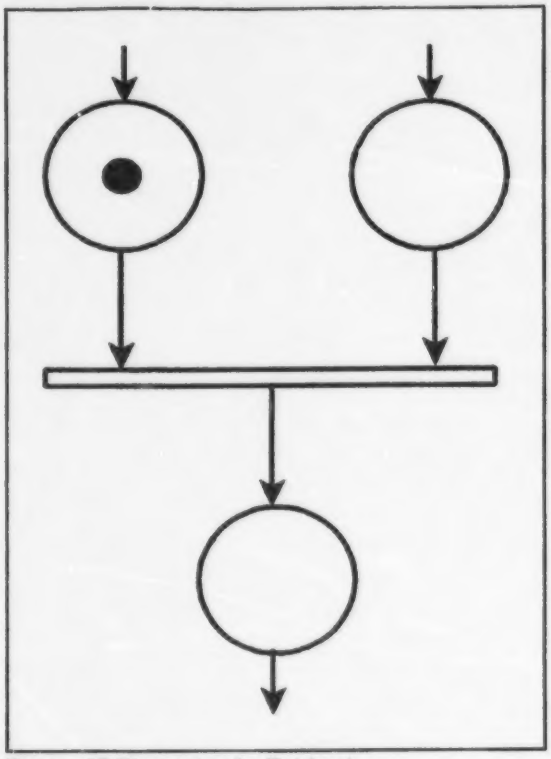

Figure 15 Elements of a Petri net.

Having set up the Petri net model, and run experiments with the model, the modeller will have two types of data available. One type consists of simple records of how the model behaved under different experimental conditions: for example, the occurrence of conflicts in a communication system. The other type of data consists of values of parameters that represent the performance of the net. Owing to its binary structure, the mathematics of Petri nets are the mathematics of Information Theory. Therefore, parameters representing the characteristics of a Petri net are fundamentally Information Theory quantities.

Beevis *et al.* provide an example of the use of a Petri net to model a conflict situation in a communication system[14]. Examples of the use of Petri nets to model command and control system analyses are reviewed in [33]. The studies reviewed are mainly ones done by Levis and his colleagues using Petri net models to explore the behaviour of command and control systems (e.g., [34], [35]). Grevet [36] used a Petri net to model a decision maker in a command and control planning process. Cothier and Levis [37], and Levis, Tomovic, and Cothier [38] used a Petri net model to study timeliness of fire in an artillery fire control plan.

Chyen and Levis [39] used a Petri net model to show that preprocessing information input to a decision maker reduces workloads when the preprocessing reduces the number of tasks the decision maker needs to do in the decision making problem. Then, after making two assumptions: first, that a preprocessor is equivalent to a decision aid, and second, that the number of tasks the decision maker does is a measure of the decision maker's workload; Chyen and Levis drew the

[14] The example is apparently drawn from the reference which they list as: Peterson, J.L. (1981). Petri net theory and the modeling of systems. New York: Prentice Hall.

general conclusion that the effectiveness of a decision aid lies in reducing the decision maker's workload.

As these examples show, Petri net formulations can be used to model a variety of command and control activities, including situations involving concurrent asynchronous events. A particular advantage of a Petri net is the capability to model the net's dynamic behaviour and provide parametric measures of behaviour - something techniques like function flow diagrams and the like cannot do. On the other hand, the underlying mathematics of Petri nets are not altogether elementary, and users of Petri net results will not likely find it easy to comprehend these results without doing a good deal of homework on the mathematics. This may limit the use of Petri net results when it comes to applying them to a real system development problem. As Levis [40] explains (apparently from firsthand experience) in discussing Petri nets' practical usefulness, establishing the validity of a mathematical model for user decision making is difficult if the user does not thoroughly understand the model, and how the model relates to the user's system.

From the HFE perspective, Petri nets can be used to model a variety of network phenomena of the sort common in command and control environments. These phenomena range from user interface and control system applications described by Beevis *et al.* [18] to the higher order command and control applications described by Levis and colleagues. For the HFE specialist, the main drawback in using a Petri net formulation will be the overhead involved in mastering the mathematics needed to use Petri net methods. Unless the specialist happens to be already familiar with Petri nets, this overhead may be considerable.

As Beevis *et al.* mention, a Petri net model soon becomes unmanageable as the size and complexity of the net increases. This suggests that Petri nets are best suited to limited (and manageable) subsystems where a Petri net formulation possibly offers the only effective means for dealing with an analysis problem.

4.2.4.3.6 Analysis of Functional Organization

Many writers, Beevis, *et al.* [18], for example, restrict the definition of function analysis to analyses of *behaviour*. However, if the analysis is an analysis of an existing system, having an analysis that shows the underlying structure that produces the system's behaviour is still useful. Some form of a block diagram is the most common way of representing a system's structure[15]. Block diagrams can show such things as:

1 the system elements that support different functions and the logical relationships among elements,

[15] As Beevis *et al.* point out, there is considerable potential for confusion when the word function is used in connection with a representation of system structure. Whatever term is used to identify an analysis of structure, the term should be clarified so that it is not confused with a function analysis of system behaviour.

2 the physical layout of the system, and the distance and communications relationships among the system's elements, and

3 the organization of the system's human elements, showing the functions and reporting relationships among units or individuals.

In command and control applications, the standard organization chart is probably the most useful type of function block diagram. An annotated organization chart is a convenient way to represent the authorities and functions of personnel in a command and control organization. Figure 16 shows an example of a functional organization chart.

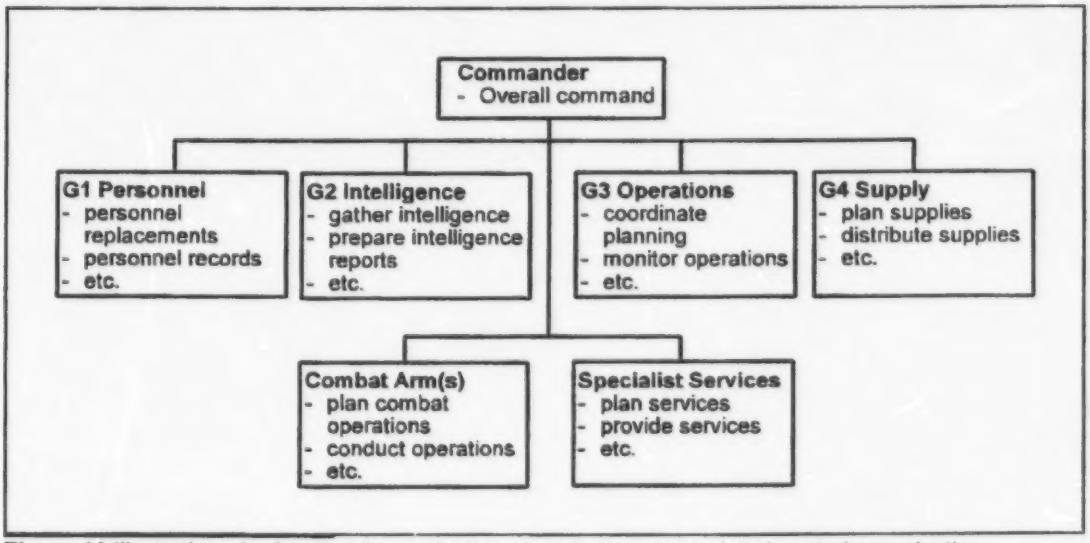

Figure 16 Illustration of a functional organization chart for a command and control organization.

The example in Figure 16 shows an army headquarters organization with its G1 Personnel, G2 Intelligence, G3 Operations, and G4 Supply functions[16], along with blocks for combat arms and specialist services. The annotations in each block are functions that can be expanded to whatever level of detail is required. Annotations can also be constructed to cross-reference function analysis data. In a multi-level command and control system, the network of hierarchical relationships is much more complicated than the organization shown in Figure 16.

Functional organization diagrams are valuable for developing an efficient and accurate mental model of a system. Having a good mental model of the system provides indispensable insight for interpreting data collected in other analyses. Later, in the design phases of development, understanding the functional organization will be critical for ensuring that the system design

[16] Although the nomenclature varies, all military combat organizations have these basic personnel, intelligence, operations and supply functions.

supports command and control tasks, and does not ignore or violate the command and control system's organizational doctrine.

Figure 17 illustrates another way of portraying a functional organization. Figure 17 shows the functional organization of a marine traffic control system in which operators (Op1 and Op2) control marine traffic [41]. The diagram shows that each operator controls traffic in a specific radar sector, relying mainly on the radar to monitor the traffic. The diagram in Figure 17 was produced as part of a project to identify technology that would improve operator efficiency.

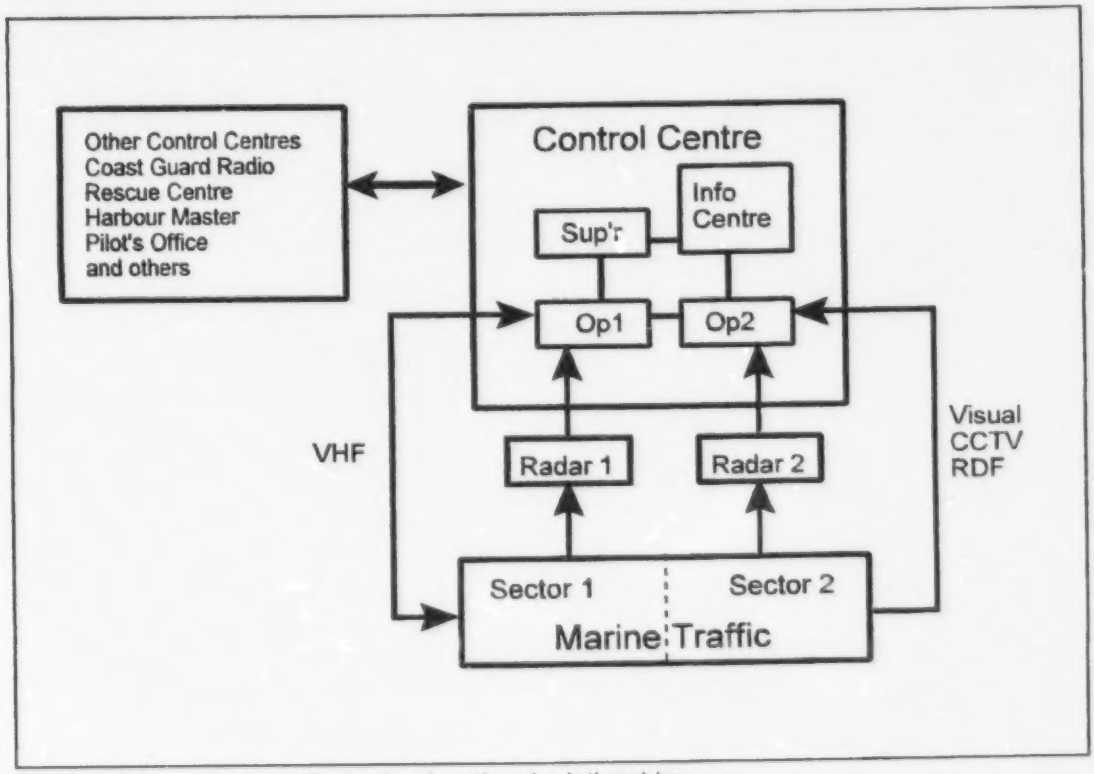

Figure 17 Block diagram illustrating functional relationships.

Although detailed analyses showed several computer-based ways to improve individual operator efficiency, the technology that would yield the greatest improvement in efficiency turned out to be evident from the functional organization diagram in Figure 17. Because each radar sector corresponds to an individual radar console and individual operator, the control centre has to staff each operator position at all times, no matter what the traffic volume. Thus, in times of low traffic volume, operators work at well below their workload capacities. The obvious solution to increasing operator efficiency turned out to be implicit in Figure 17. This solution was to have a synthesized all-sector radar display that could be divided among operator positions depending on the traffic volume. So, for example, during low traffic volume, the entire synthesized display

could be directed to one operator position, and allow this one operator to handle all the traffic control centre's traffic. During high volume traffic, the traffic could be divided between operators in the manner corresponding to that shown in Figure 17.

In summary, functional organization diagrams are useful for getting an overview of a system, and on occasion, may be useful for identifying ways to improve a system design.

4.2.4.3.7 Other Function Analysis Methods

The function analysis methods described here make only a sample of the large number of function analysis methods that can be found in the literature. The methods just described typify available methods, and in particular cover the two main classes of methods: 1) function flow methods, and 2) network methods like the Petri net. Other methods such as State transition Diagrams and Behaviour Graphs are described in Beevis *et al.* [18]. More methods can be found in various system engineering textbooks.

Besides other published methods, many contractor firms have their own system development tools which include function analysis tools. In some cases, these tools represent a purely internal set of working practices. In other cases, the tools may be trademarked products available for use under license, SADT™ [30], for example. Method/1 is another example of a commonly used proprietary system development tool [42]. Proprietary methods range from those that represent specific technical methods (like SADT™) to those that are a combination of technical methods and system development management tools (like Method/1).

Usually, proprietary development tools provide convenience in the form of standardized technical and management procedures and documentation. However, they seldom, if ever, put constraints on the technical methods that system development personnel can apply to an analysis problem. Moreover, any technical methods specified in a proprietary tool will normally be derived from established methods described in open technical literature.

4.2.4.4 Function Analysis and Phases of the Command and Control Cycle

The earlier discussion of mission analysis showed that the different characteristics of the planning, directing and monitoring phases in the command and control cycle made it reasonable to use different mission analysis approaches to describe the different phases. Extending this reasoning to function analysis suggests that different function analysis methods might be used for different command and control phases.

Since planning consists of a standard sequence of coordinated activities, a function flow approach is an obvious choice for analysing planning. Monitoring also contains fixed sequences of activities done in response to the various types of messages handled by a headquarters. However, the activity sequences in monitoring are generally simple, mechanical, and of short duration. While a function flow analysis of monitoring activities will be instructive, it will not capture what is

probably the more important aspect of monitoring activities: the demands for service imposed by the asynchronous flow of message traffic during monitoring. A network function analysis method is the obvious choice for analysing functions associated with message traffic in monitoring.

A network analysis, not only for monitoring but also for planning, is also an obvious function analysis choice from a pure CCIS perspective. As discussed earlier (see Section 4.2.4.2.4, page 53), a CCIS is fundamentally a data base and communications network, and therefore a natural candidate for a network-based function analysis. From the command and control examples used in previous sections to illustrate different function analysis methods, it will be evident that a pure network function analysis may miss important features of command and control operations. Therefore, the analysis of monitoring activities may require both network analysis methods and other methods like function flow analysis.

4.2.4.5 Data Gathering Procedures for Function Analysis

Ideally, the function analysis information needed to satisfy the HFE requirements will come from the system engineering function analysis, which is normally done by non-HFE personnel. However, as explained in the previous section, and earlier in Section 4.2.4.2.4, (page 53), some function analysis methods are better suited to HFE needs than others. Therefore, the extent to which the general system engineering function analysis results will meet HFE needs will depend on what function analysis methods were used.

As the section following this one explains, the CCIS HFE Requirements specifies function analyses describing four attributes: 1) the organizational structure of the command and control organization, 2) the command and control system's information flows, 3) its information management processes, and 4) its decision making processes. If the system engineering function analysis includes this information, little HFE effort will be needed to get the results into shape for HFE applications. If the system engineering analysis does not provide a basis for the function descriptions required in the CCIS HFE Requirements, the analyses must be done separately.

If it becomes necessary to gather separate function analysis information for HFE purposes, the most efficient strategy will likely be to gather this information as part of the task analysis effort described in Section 4.2.5, (page 73). Section 4.2.5 describes how to set up a task analysis data base, and how to collect task analysis data. Although specifically addressed to task analysis, with a little modification, the procedures in Section 4.2.5 apply equally well to doing a function analysis.

4.2.4.6 Summary of Function Analysis Methods

The following summarises the application of function analysis methods in CCIS development.

1 Function analysis is an analysis process that decomposes a system's overall function into the hierarchy of sub-functions needed to support the overall function.

2 A function analysis can be done either to identify ways to improve the function of an existing system, or to identify functional requirements for a new system.

3 Although function analysis is normally done by non-HFE system engineering personnel, function analysis results support several critical HFE analysis and design functions. These include:

3.1 providing an overall understanding of the system's operation,

3.2 providing data for the user task analysis,

3.3 relating HFE design recommendations to system functional requirements, and

3.4 defining measures of user performance and system performance.

4 There are many function analysis techniques available. They fall into two main classes: 1) function flow techniques to describe the flows of activities from a system's inputs through various system functions to the system's outputs, and 2) network techniques to describe the interactions among a system's sub-functions. In phases of the command and control cycle, function flow techniques are best suited to describing planning activities and the procedural aspects of monitoring activities. Network methods are best suited to describing the communications aspects of monitoring activities.

5 System engineering function analysis methods vary in the degree to which their results support HFE function analysis needs. If the function analysis results do not support the HFE function analysis requirements specified in the CCIS HFE Requirements, extra functional analyses will be needed to support these requirements. Usually, the task analysis effort will provide the most efficient vehicle for filling gaps in function analysis information.

4.2.5 Task Analysis

This section will describe 1) the purposes of task analysis in system development, 2) the CCIS HFE Requirements task analysis requirements, 3) how to set up a task analysis data base, and 4) methods for collecting task analysis data. Most of the contents of this section aim at giving practical advice on how to do a task analysis. The advice is explicitly concerned with the type of task analysis specified in the CCIS HFE Requirements. However, almost everything in this section can be applied to conducting any HFE analysis, including a function analysis.

The objective of a task analysis is to describe, as completely as necessary for the purposes of the analysis, the tasks that humans do in a system to make a system work. As this objective implies, the human behaviour described in a task analysis depends on the purpose of the analysis. In the CCIS HFE Requirements, the fundamental purpose of task analysis is to support all the HFE activities that influence the development of the system and its implementation. Thus, nearly all the requirements in the CCIS HFE Requirements are premised on having a comprehensive task analysis data base that serves as the backbone of the HFE plan throughout the system development plan. Table 3 outlines the relationship between task analysis activities and two other

Table 3 Relationship among system development activities, task analysis activities, and other HFE activities in each phase of system development.

System Development Activities	Tasks Analysis Activities	Other HFE Activities Dependent on Task Analysis
Analyse Requirements – Perform mission analysis – Perform function analysis – Identify CCIS requirements	– HFE mission & function analysis statements – Task analysis of current command and control operations	
Design System – Perform function allocation – Design system	– Update task analysis to reflect user tasks in emerging system design	– Contribute to function allocation – Advise on HFE aspects of system design – Conduct user tests of design options
Test System Design – Conduct hardware and software tests – Conduct user tests		– Conduct user tests of system design – Identify design changes needed to correct inadequate user task performance
Revise System Design – Produce final system design based on results of design tests	– Revise task analysis – Produce task analysis of users' tasks in final system design	– Conduct any user tests needed to validate design revisions
Produce/Implement System		– Develop or contribute to user training procedures, and personnel policies and procedures

sets of activities: 1) other system development activities, and 2) other HFE system development activities that depend on task analysis data.

The first column of Table 3 lists the system development activities in the five main system development phases described earlier in Section 3.2 (page 12). The second column of the table lists the task analysis activities associated with each system development phase. The third column of the table lists the other HFE activities associated with each system development phase. The task analysis column of the table shows how the task analysis data base evolves during the system development process. At the beginning, the task analysis describes users' tasks in the predecessor system. As development progresses, the task analysis is continually updated so that it always describes user tasks in the emerging system design. The task analysis information can then provide the basis for all the other HFE activities that influence system design. Figure 18 shows the cyclical relationship among system development, task analysis, and HFE design and testing activities.

Figure 18 shows a cycle in which user task descriptions provided by task analysis activities determine the nature of the HFE design and testing activities that influence the system design. The position of task analysis activities in the cycle makes them critical to the effectiveness of the HFE plan. This is because HFE design and testing activities cannot be done effectively without a thorough knowledge of users' tasks. For example, it is impossible to do intelligent HFE design reviews, or provide prescriptive design advice from HFE design reviews, without knowing the tasks that users are to perform with the system. Devising user performance measures for HFE design tests without knowing users' operating tasks is similarly impossible. Deriving useful design advice from design test results that do not directly reflect users' operating tasks is equally impossible.

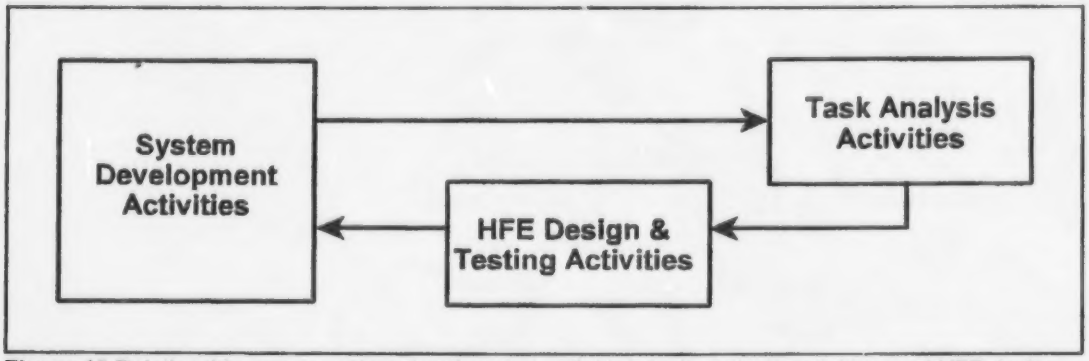

Figure 18 Relationship among system development activities, task analysis activities, and HFE design and testing activities.

The task analysis perspective just outlined implies that, for the purposes of the CCIS HFE Requirements, task analysis has a dual role in the application of the CCIS HFE Requirements to a CCIS development. These roles are:

1 to provide a comprehensive and perpetual data base of information about users' tasks in the developing system, and

2 to serve as a tool for studying user behaviour and testing user performance capabilities.

The following sections on task analysis develop these roles in four subsections that describe:

1 the CCIS HFE Requirements specifications of the information to be included in the task analysis data base;

2 ways to organize the task analysis data base and keep it current with the development of the system design;

3 the use of HFE task analysis technical methods for CCIS research and testing applications, and

4 methods for collecting task analysis information.

4.2.5.1 CCIS HFE Requirements Specifications for Task Analysis

The following is divided into two sections. The first describes the CCIS HFE Requirements' general specifications for task analysis. The second describes the CCIS HFE Requirements specifications of the types of information to be collected in a task analysis.

4.2.5.1.1 General Task Analysis Specifications

CCIS HFE Requirements (4.1.1) specifies analysing users' tasks as part of the baseline analysis that will be used to identify the CCIS design requirements. CCIS HFE Requirements (5.1.1.4) describes the detailed task analysis requirements that make up the initial system analysis activities in the system development project. These task analysis requirements are part of the overall HFE analyses requirements described in CCIS HFE Requirements (5.1.1). The CCIS HFE Requirements also requires a task analysis describing users' tasks in the system design, and requires that this analysis be kept current with the system design. CCIS HFE Requirements (5.2.1) describes these requirements.

The DID titled Human Factors Engineering Plan specifies how to report plans for conducting task analysis work. The DID titled Human Factors Engineering Analysis Report gives specifications for documenting tasks analysis data, and reporting on task analysis work.

Besides the CCIS HFE Requirements sections and DIDs that specify doing and documenting task analysis work, most of the other requirements and DIDs have indirect references to task analysis data. These indirect references reflect the fact that HFE analysis data are the foundation for nearly all the HFE activities that go into CCIS design and testing.

The CCIS HFE Requirements (5.1.2) specifies that the results of the function and task analyses are to be further analysed to identify measures of user and command and control system performance and effectiveness. Methods for meeting this requirement are described in Section 4.2.7 (page 121).

4.2.5.1.2 Task Analysis Data Specifications

The CCIS HFE Requirements (5.1.1.4) specifies a task analysis that includes six types of information to be included in the description of each task:

1 the information required to initiate and perform the task,

2 the information available when the task is performed,

3 the cognitive nature of the evaluation process used to decide what action to perform in response to task initiating information, and an estimate of the mental workload imposed by the evaluation and decision process,

4 the decision that will be taken after the evaluation process,

5 the action taken to perform the task, and

6 the nature of the feedback indicating the adequacy of action taken.

The CCIS HFE Requirements (5.1.1.4) further specifies that related groups of tasks performed to carry out discrete system functions are to be analysed to describe:

1 task sequence and simultaneity characteristics,

2 timing and accuracy requirements,

3 criticality for mission accomplishment,

4 frequency of task performance,

5 proportion of available user time required to do the tasks,

6 interactions with other personnel,

7 communications required in doing the tasks,

8 any special hazards involved in the tasks,

9 procedures, references and equipment required to do the tasks, and

10 skill, training and experience needed to do the tasks.

In the following subsections, these 10 task information items will be called supplementary information items.

The CCIS HFE Requirements specifications represent what amounts to a task inventory. For an individual task, the task description shows what is involved in doing the task, along with information useful for a variety of special purposes such as identifying training requirements, identifying operating hazards, and so on.

At first sight, the task description information specified in the CCIS HFE Requirements appears to be a lot of information to collect. Keep in mind however, that the supplementary information items will be the same for groups of related tasks: for example, all the sub-tasks making up a particular major task. Consequently, when the task analysis data are collected, supplying the supplementary information for every individual sub-task will not be necessary. Usually, the information for a first level task or second level sub-task will also cover all the subordinate sub-tasks. Because of this, the task analysis data collection effort will be much smaller than the number of information items might suggest.

The CCIS HFE Requirements does not explicitly define what a task is; nor does it specify how to represent sequence and branching relationships among tasks. The next section discusses possible approaches to defining tasks and expressing relationships among tasks.

4.2.5.2 Structuring Task Analysis Data

The main objective of the task analysis approach specified in the CCIS HFE Requirements is to describe users' tasks in relation to system operations, in particular, the users' interactions with the command and control system during command and control operations. This reflects a task analysis approach that is fairly traditional in the literature. It is usually called critical task analysis (e.g., Beevis *et al.* [18]). In this tradition, Miller [43], for example, defines a task by reference to five properties that represent the minimum information needed to describe a task. The properties are:

1 the indicator that is the source of information for signifying that the task is to be started,

2 the indicator state that signifies that the task should be performed,

3 a system object or control acted upon when the task is performed,

4 a description of the action performed on the system object,

5 the feedback that indicates the adequacy of the task action.

Taking the same approach, Drury, *et al.* [44], define a task as a set of human actions that contributes to achieving a system objective[17]. Drury *et al.* describe tasks according to three attributes:

1 the perceptions, decisions, and control actions that go into doing the task;

2 the system event that initiates the task, and

3 the system state that represents achievement of the task objective.

Except for not including explicit references to the system objects of task actions, Drury *et al.*'s task properties are essentially the same as Miller's. Although different authors in different contexts provide variations on the definitions given by Miller, and Drury, *et al.* (for example, Meister [5], Laughery & Laughery [25]), overall, the essence of any task analysis is to describe an individual's behaviour in interacting with a system.

Drury, *et al.* extend the definition of a task by defining three types of tasks: discrete tasks, continuous tasks, and branching tasks. A discrete task is a discrete set actions performed in a fixed sequence, and usually in a fixed time. A continuous task is one or more continuous actions performed over an extended time. A monitoring task such as keeping watch on a system status display is an example of a continuous task. Finally, a branching task is the act of choosing one of several alternative actions in response to a system event. Another way of conceptualizing Drury *et al.* task types is to define just two task types, discrete or continuous, and then subdivide each type into being either branching or non branching. Having branching tasks is not so much a matter of defining a unique 'branching' characteristic for tasks, as it is a matter of being able to portray the different sequences in which tasks may be performed during operations.

[17] This task definition is also used in [17].

The definitions just outlined are consistent with six task description items specified in the CCIS HFE Requirements, and just described in Section 4.2.5.1 (page 76). The additional ten task information items that the CCIS HFE Requirements specifies for related groups of tasks (see page 77) are not commonly considered essential in any particular task analysis. The ones specified in the CCIS HFE Requirements are items that would conceivably be useful in CCIS environments. However, in a particular CCIS development project, all the items may not be needed. Conversely, additional or different items might prove useful. Thus, deciding what supplementary task information items to include in the task analysis can be treated as a matter of HFE Plan tailoring as specified in CCIS HFE Requirements (6.4). In principle, the amount and variety of information that one might attach to a task description is almost limitless. However, at a minimum, any task description scheme must describe the individual user's behaviour, how this behaviour is influenced by the system, and how the user's behaviour influences the system's behaviour.

The task conception provided by Drury *et al.* provides a practical basis for describing relationships among tasks, particularly sequential relationships, branching relationships, and division of tasks into discrete and continuous types. All that needs to be added is a provision for decomposing tasks into hierarchies of sub-tasks.

Defining a hierarchy for a task analysis is no different from defining a function hierarchy for a function analysis. However, there are no formal rules for how to decompose a task into sub-tasks. Meister [5] suggests that choosing how to decompose a particular task into sub-tasks is a matter of applying knowledge about human behaviour to the task performance requirements imposed by the system. In practice, few analysts have difficulty deciding just how to break a task into subtasks.

Besides deciding how to break down a task, the analyst also has to decide how many levels of sub-tasks will be needed to describe task performance adequately. Again, there are no generally accepted rules for determining how many levels will be needed in the task hierarchy. Sometimes a logical rule can be defined for a specific task analysis. Annett and Duncan [45], for example, developed a rule for deciding when a task had been analysed in enough detail to identify the task's vulnerability to operator error. Usually, any rule for deciding the number of levels of analysis needed will be a rule derived from the purpose of the task analysis. For example, a task analysis done to identify general user training needs typically requires little detail, and so, few levels of task breakdowns. In contrast, a task analysis done for the design of a critical operating procedure may require breaking tasks down into considerable detail. These examples suggest that during a development project, the level of detail required in the task analysis data base will vary with the phase of development and the uses to which the task analysis data are put. This further suggests that the overall task analysis approach should be designed to be adaptable to the evolving demands of both the development process as a whole, and the demands of the different HFE activities that are supported by the task analysis data. The next section outlines some possible approaches to setting up an efficient task analysis data base.

4.2.5.3 Setting Up a Task Analysis Data Base

A task analysis data base has five basic requirements:

1 a logical data structure corresponding to the task hierarchy and the task description items defined for describing tasks;

2 a physical data storage medium that:

- 2.1 is easy to set up;
- 2.2 provides convenient means for entering the task description data into the database;
- 2.3 has facilities for doing analyses of the task data: statistical analyses, for example, and
- 2.4 has facilities for selecting, manipulating and displaying the task analysis data.

The next two subsections suggest ways to meet these requirements. The first describes a procedure for defining the logical structure of the database. The second describes possible physical media for storing the task analysis data, and compares their facilities for setting up, entering data, and analysing data.

4.2.5.3.1 Task Analysis Database Logical Structure

The following suggests the steps to take to set up the structure of the task analysis data base.

1 Define a basic database record such that there will be a record for each task in the *lowest level of the task hierarchy*.

2 Define a task index numbering system that uniquely identifies each task, the task's position in the task hierarchy, and the task's sequential position relative to other tasks that are normally performed before and after it.

3 For each record (i.e., task), identify a common set of data fields that include:

- 3.1 fields corresponding to the items of information that describe the task and its properties,
- 3.2 a field containing the task index number,
- 3.3 a field containing the index numbers of tasks to which the task can branch, and
- 3.4 fields for any additional information that might be useful to include in the database.

4 For each record data field, define:

- 4.1 the name of the field,
- 4.2 the field's data format,
- 4.3 for text information, any syntax that the field's contents must conform to,
- 4.4 the maximum size of the field or the range of values that the field may contain, and
- 4.5 any codes needed to indicate such things as missing values.

Defining a record corresponding to the lowest level of the task hierarchy ensures that complete information can be retrieved for every task no matter what the task's level in the hierarchy. At the same time, the structure allows retrieving information only for tasks at or above a certain level in the task hierarchy. An alternative record structure is to create a relational database in which records for high level tasks are parents for other sets of records representing sub-tasks. However, since the information fields for every task record is the same, no matter what the level of the task, the relational database provides no real advantage, and only increases the complications of setting up and managing the database.

A legal style numbering system (the same as the system used to number sections in this document) will provide an adequate task numbering system. In this system, the top level numbers represent major tasks, second level numbers represent second level sub-tasks, and so on; thus:

1 Major task 1
  1.2 Sub-task 2 of major task 1
    1.2.1 Sub-task 1 of the sub-task 2 of major task 1.

If task information is recorded so that the sequence of task numbers corresponds to the order in which the tasks are normally performed, then the magnitudes of the task numbers will also represent the sequence in which the tasks are performed.

Defining the fields for the task information is a matter of making a list of the information items to be recorded for each task. For example, the task information requirements in CCIS HFE Requirements (5.1.1.4) provide a basic list of fields for the task analysis database. In addition, another field will be needed for the task index numbers, and another field will be needed for index numbers of tasks to which the task can branch.

Besides the items specified in the CCIS HFE Requirements, the task description should include a field that gives a succinct description of what the person who performs the task actually does to perform the task. The task descriptors in the CCIS HFE Requirements, in common with most task analysis schemes, describe the characteristics of tasks without directly describing how tasks are performed. A description of how each task is performed will make the task analysis data more comprehensible to people who need to apply the data to other development activities.

Once the task analysis data fields have been defined, the next step is to assign a name to each field, and define the field's data format. Since most presentations of task analysis data will likely be as tables, the name of each field should be a short title suitable for a table heading. Each field's data format will follow from the nature of the field's data: a text format for text or narrative data, and a numeric format for data that either is native numeric data or can be coded as numeric data. Most task analysis information can be recorded as numeric data or numeric codes. For example, most of the ten supplementary information items required by the CCIS HFE Requirements (see the list on page 77) can be recorded as numeric codes. Using numeric codes permits doing tabulations and statistical analyses of the task data, and saves physical storage space.

Beyond defining the basic data format, the analyst also has to specify rules defining valid field content. For text fields, which will include narrative descriptions of the tasks, the field content rules should specify the syntax to be used in constructing the task description. For example, suppose that the task description is set up with six fields representing the items specified in the CCIS HFE Design Guidelines, that is, 1) the information required to initiate and perform the task, 2) the information available when the task is performed, 3) the cognitive nature of the evaluation process used to decide what action to perform in response to task initiating information, and an estimate of the mental workload imposed by the evaluation and decision process, 4) the decision that will be taken after the evaluation process, 5) the action taken to perform the task, and 6) the nature of the feedback indicating the adequacy of action taken. Each field definition should have a rule specifying the construction of the narrative that conveys the meaning of the information item. Although it may occasionally produce cumbersome expressions, specifying a conventional subject-verb-object construction ensures that the task descriptions will be comprehensible when they are read long after the analysis was done, or are read by non-HFE users of the analysis.

Another rule or coding scheme that can be useful is one that shows the scope of the information contained in a field for a task or sub-task record. It was mentioned earlier (Section 4.2.5.1, page 76) that the information items for groups of tasks will often be the same for all the tasks in the group. For example, if a task is critical for mission accomplishment, then all its subordinate sub-tasks will be critical for mission accomplishment. So, designating the task as critical to mission accomplishment implies the mission accomplishment codes for all the task's subordinate sub-tasks. Such a rule would probably not work for an item like designating special hazards involved in a task.

A special hazard would typically be connected with a specific sub-task. Here, the superordinate task as a whole could be considered hazardous, but only because of one of its sub-tasks. Thus, a task designated as hazardous would not necessarily mean that all of its subordinate sub-tasks are hazardous. Nor would the designation be much help in identifying the hazardous sub-task. Here a reasonable rule might be that a task coded as hazardous means that at least one sub-task is hazardous, and the hazardous sub-tasks are specifically the ones designated as such. These examples illustrate that setting up task information scope rules can simplify and reduce database storage space. They also illustrate that the rules have to be defined separately (and carefully) for each task information item.

The final step in setting up the task analysis database is to define the maximum size of each data field or the range of values that the field may contain, and any codes needed to indicate such things as missing data. This last step completes the specifications for setting up a medium for recording and storing the task analysis data.

4.2.5.3.2 Media for Recording and Storing Task Analysis Data

If the task analysis database logical structure meets all the requirements in the previous section, almost any medium can be used to record and store the task analysis data. The basic media options for storing task analysis data are:

1 a manual spreadsheet,

2 structured computer text file,

3 computer spreadsheet program,

4 computer database program, and

5 custom task analysis computer program.

The following describes each medium and its advantages and disadvantages.

Manual Spreadsheet

A manual spreadsheet can be set up on any suitable paper medium with columns to represent the task description data fields. It can be then filled in as the task data are collected. Although most analysts will eventually want to get the task analysis data into a computer medium, the manual medium has some advantages that should not be overlooked, especially for experimenting with the database setup, and testing the data collection procedure. A manual spreadsheet can be set up with very little effort compared with setting up a computer-based medium. This makes the manual spreadsheet an efficient way to pilot-test the task analysis setup. Entering some collected task analysis data into the sheet will soon show if the setup will work. Once pilot-testing shows what would be the best setup, a computer-based medium can be created.

A manual spreadsheet can be an efficient device for recording task descriptions during data collection from subject matter experts and other data sources. Compared to taking freestyle written notes, the manual table ensures that the analyst collects all the required data for each task. When data are collected form subject matter experts, the analyst can immediately check the accuracy of the data with the subject matter expert.

Computer Text File

A task analysis database can be set up as a text file by defining each line in the file as a record containing the data for a designated set of task description data fields. Within each record, fixed column positions and numbers of columns are allocated to the different data fields. The file can be set up so that each record (i.e., line) in the file contains the data for a task. If the overall task description has many data fields, the fields can be divided among several consecutive records.

A text file has several advantages over more sophisticated computer media such as spreadsheet and database files. It can be set up almost as easily as a manual spreadsheet, so it is a good way to pilot-test a setup. A text file can be maintained and edited directly with a text editor or a word processing program, so file troubleshooting is easier than it is with spreadsheet or data base files. A word processing program, with its macro programming and tables facilities can also be used to manipulate the data and create various presentations of the task data. Finally, a text file can be read by almost any computer application. For example, most advanced statistical analysis programs use text files as their data input files. In addition, since spreadsheet and database

applications can read text files, all or parts of a text file data base can be converted to a spreadsheet or database whenever the analyst wants.

The main disadvantage of a text file is its lack of a convenient input medium. To enter data into the file, the data have to be typed directly into the file in the exact format defined for the file. About the only way to do this reasonably accurately is to have the data already hand recorded on some type of fixed data entry form - a procedure that will involve considerable manual labour. An alternative data entry approach is to write a computer program that provides a data input form and converts the input to the required text file format (see Custom Task Analysis Program below).

Computer Spreadsheet

Since a computer spreadsheet is a computer rendition of a manual spreadsheet, it will, in principle, serve as well as a manual spreadsheet for recording task analysis data. Computer spreadsheets are easy to set up. They support doing a variety of manipulations and calculations on the task data. They also support displaying and printing the data in a variety of formats. Spreadsheets are not as good as manual spreadsheets for displaying all the data for an individual task. With a manual spreadsheet, one can easily see all the data for a task all at once. In contrast, a computer spreadsheet can typically only show data a few columns at a time.

Computer spreadsheets provide better data input facilities than text files permit. Although most spreadsheet programs support setting up a data input form, setting up the form usually requires a fair amount of work, and thorough knowledge of the program's operation.

Database Program

A database program is a natural choice for storing task analysis data on a computer. With a database program, the analyst can create a data input form for entering task data. The analyst can also create a variety of forms and tables for displaying either the data for an individual task, or tabulations of data for groups of tasks. Database programs also have powerful enquiry facilities for finding and selecting records that meet specified selection criteria. Most database programs can also support a relational task analysis structure in which there is not only a hierarchy of tasks, but also a hierarchy of task structures.

Setting up a database program with its various forms and tables takes a good deal of work. Database files cannot be edited directly, and can be difficult to repair if they become corrupted. Unless one is used to database programming, the database program may not offer much advantage over a more familiar spreadsheet program. Data base programs are designed for multiple users doing simultaneous transactions on the same database. For an effectively single-user database application, which a task analysis application is usually, the unneeded multiple user facilities may create nuisances without which the analyst could do.

Custom Task Analysis Program

If one can write programs in a high level programming language, creating a program for collecting or entering task analysis data may be worthwhile. The most labour intensive part of a task analysis, or any type of function analysis for that matter, is collecting and entering the analysis information. A custom program permits one to create efficient input forms with such things as automatic indexing of tasks, drop-down lists for coding numerically coded task properties, and facilities for dealing with the hierarchical properties of tasks. Having a convenient way of dealing with these features of task analysis can save considerable labour in task analysis data collection activities - convenience that spreadsheet and database programs do not provide.

A custom program with more than elementary display and editing facilities would probably not offer any advantage over a spreadsheet or database program. Rather than reinventing spreadsheet and database capabilities in a custom program, having the custom program able to generate files that could be read into spreadsheet and database programs will likely be more efficient.

There is no publically available software specifically designed for task analysis applications. Because of the wide variety of task analysis methods, developing a task analysis program that could deal with all the methods would be difficult. The database program is the nearest thing to what could be considered a general purpose task analysis program.

Summary of Task Analysis Database Media

Table 4 summarizes the advantages and disadvantages of the different task analysis database media.

The first column of the table lists five features that gauge a medium's convenience of use. Taken together, the features represent the overall amount of effort that will be needed to use each medium in a task analysis. This choice of features comes from putting a priority on minimizing the effort required by the task analysis. It also implies that none of the media have any outstanding theoretical or technical advantages.

The five rightmost table columns show ratings of the different media on each of the seven ease-of-use features listed in the first column of the table. The ratings are distilled from the contents of the preceding discussion of the merits of different possible task analysis database media. The low, medium and high ratings are intended as ratings of each medium compared with the other media, rather than as absolute ratings. The ratings also presume that the medium user has no more than ordinary experience at using each medium. Conversely, a user who is expert at using a particular medium would likely judge that medium superior to the others.

The ratings in Table 4 suggest that while computer-based media take more effort to set up, they are superior to the manual spreadsheet for data analysis and for producing reports. Across computer-based media, probably no one medium is particularly superior. Again, from the

Table 4 Comparison of task analysis data base media based on low, medium and high ratings of media on convenience of use features.

	Manual Spreadsheet	Text Data File	Computer Spreadsheet	Computer Database	Custom Program
Effort to learn medium	Low	Low	Medium	High	High
Effort to set up	Low	Low	Medium	High	High
Effort to input data	Medium	High	Medium	Medium	Low
Effort to maintain files	Low	Low	Medium	High	Low
Flexibility for producing reports	Low	High	Medium	Medium	High
Effort to produce reports	High	Medium	Medium	Medium	Medium
Flexibility for data analysis	Low	High	Medium	Medium	High

individual analyst's perspective, the best medium is the one that he or she is more most comfortable with.

4.2.5.4 Procedures for Collecting Task Analysis Data

In a CCIS project, task analysis data gathering occurs in two different circumstances, each requiring a different approach. The first circumstance comes at the beginning of a project when tasks in the existing command and control system are analysed. The second comes in the system design phases when the task analysis is done to document users' tasks in the emerging system design. The following subsections describe practical task analysis procedures for these two circumstances.

4.2.5.4.1 Task Analysis Procedure for An Existing System

The following steps provide an efficient procedure for collecting user task descriptions for an existing command and control system.

1 Identify sources of task information.

2 Find and review the documentation.

3 Plan the data collection procedure.

4 Construct data gathering instruments.

5 Recruit the subject matter experts.

6 Interview and debrief the subject matter experts.

The next six subsections describe these steps.

4.2.5.4.1.1 Identify Sources of Task Information

Subject matter experts and system documentation are normally the two basic sources for task analysis information.

Subject matter experts include experienced users who do the tasks, supervisors who are in charge of the users who do the tasks, and training personnel who train the users. Other subject matter experts may include technical specialists who support users' operations, or who have conducted prior studies of the users' operations. Normally, users who do the tasks will be the primary source of subject matter expertise, with other personnel serving to validate task descriptions, or answer task-related technical questions.

Documentation includes standard operating procedure documentation such as operations manuals, staff college and specialist schools training materials, any mission and functional analysis reports already done for the development project, and reports from past technical studies. One should not forget to ask if an existing task analysis is available. Although a current task analysis is seldom available, one done some years past may be. Even if an old analysis is considerably out of date, it can provide useful background, and be useful for planning a new analysis.

The protocol for getting access to documentation and subject matter experts will depend on the project organization. However, the PM is always ultimately responsible for providing this access.

4.2.5.4.1.2 Acquire and Review Documentation

Quickly review the items of documentation to sort the items into three categories: 1) items that will directly contribute to producing actual task descriptions, 2) items that will serve as references on matters such as doctrine, and 3) items that can be put aside as unlikely to be of use. Then review the items in the first category to identify major functions or task groups, and to create a preliminary task analysis database structure that identifies the major tasks and possibly some sub-task groups. In reviewing the documentation, identifying areas of documentation that will need revision when the new system is implemented may also be useful.

4.2.5.4.1.3 Plan the Data Collection Procedure

Collecting task analysis data generally consists of some combination of three activities: 1) getting task descriptions from subject matter experts, 2) observing tasks being performed, and

3) extracting information from documentation. How these activities should be arranged for a particular analysis depends on the nature of the system being analysed and the analyst's preferred working methods. The following steps make up a data gathering plan that would suit most situations.

1 Review documentation and identify major functions and tasks.

2 Observe operations to get a feel for the operating environment.

3 Work with subject matter experts to get initial narrative descriptions of the tasks.

4 Verify task descriptions against documentation and identify discrepancies between operations practice and documented procedures.

5 Have independent subject matter experts verify the accuracy of the narrative task descriptions.

6 Work with subject matter experts to complete supplementary task description items.

7 Have independent subject matter experts verify the accuracy of the completed analysis.

8 Observe operations to verify the task analysis results.

The analysts should get as much information from documentation as possible before starting work with subject matter experts. This will avoid wasting subject matter expert time on things that could have been learned from documentation. And the more prior knowledge the analyst has, the easier it will be for the analyst to get on common ground with the subject matter experts when the analysis starts.

It is worth observing operations at the beginning of the analysis to get oriented to the operating environment. As a rule, direct observation is not a practical way of gathering task analysis data. Exceptions to the rule are situations where the user's workload is low enough to allow the user to explain what he or she is doing while doing tasks, or where a subject matter expert can explain what the user is doing to perform a task. Even when it possible to gather information from observing operations, the analyst has to analyse tasks as they occur, which may not be in the most convenient order for doing the analysis. Also, the analyst may have to wait through many repetitions of frequently performed tasks until rarely performed tasks come up, or never get to see rarely performed tasks at all.

Video tapes of operations may occasionally be useful for getting task analysis information. However, video tapes must be viewed to be analysed. According to Nielsen [46], analysing a video tape may take three to 10 times longer than the duration of the tape. Since a subject matter expert will usually be needed to explain what is on the tape, the time needed to analyse a tape will usually be better spent getting task descriptions directly from the subject matter expert. Although video tapes are usually too inefficient to be worthwhile, still photographs can provide an efficient form of note-taking when observing operations. Photographs provide detailed records of equipment and facilities, and generally good reminders of what the operations environment is like.

Working with subject matter experts to get task descriptions is usually best done away from operations. Collecting task data in two stages is also best: the first stage to get the narrative

descriptions, and the second stage to pick up supplementary task information. Having the subject matter expert provide all information for each task at once breaks up the subject matter expert's train of thought and slows the pace of the work. Moreover, since much of the supplementary information will be the same across groups of tasks, collecting the supplementary information will be more efficient if the task groups are identified beforehand.

Subject matter experts need a concrete framework to guide and organize their production of task descriptions. The mission and functional analyses' results can usually provide this framework. It should also be possible to go a step further and create a framework consisting of a preliminary high-level task analysis derived from the mission and function analysis results. Collecting task data from the subject matter experts will then be a matter of expanding and elaborating the initial task analysis.

Verifying task descriptions against documentation can be done in either of two ways: 1) by checking documentation as the task descriptions are collected, or 2) by checking documentation after the narrative descriptions are completed. Normally, the subject matter expert will know the documentation, and may in fact refer to the documentation to produce the task descriptions. The subject matter expert will usually know of any discrepancies between documentation and practice, so discrepancies can be recorded as the analysis goes along. In cases where documentation is seldom referred to in operations, checking the task descriptions against documentation in a separate exercise will be necessary.

Task descriptions should be verified by independent subject matter experts. If the narrative descriptions of the tasks are passably readable, the independent review can be done by sending the descriptions to the subject matter experts and asking for opinion: either in written comments or in a face-to-face meeting. It is a matter of judgment whether to have the independent review done in two stages: once for the task descriptions, and once for the completed analysis. If the review is to be done in one stage, reviewers should be told to go through the narrative descriptions first, and then to review the supplementary information. If the analysis is long, consider dividing the analysis into shorter sections less likely to overwhelm the reviewers.

Special care needs to be taken in the use of subject matter experts for analysing high level cognitive tasks such as decision making and tactical planning. Procedures for doing high level cognitive tasks are often not documented, and are done in different ways by different individuals. Since it is not the analyst's job to rationalize the user organization's procedures, nor is it the subject matter experts' job to simplify the analyst's life, it may be necessary to record alternative descriptions for a task.

At the completion of the task analysis, arrange to observe operations to compare the task descriptions with what users do in operations. Use the observations to confirm the accuracy of the task analysis. Observing operations will also help to confirm or identify important information that the task analysis could not capture. For example, a task analysis will not show many of the sequence and timing characteristics of tasks, nor will it show the statistical properties of task performance. Observing operations in light of the completed task analysis will help to define the analyses needed to get this additional information.

In summary, the plan just outlined contains all the elements that go into collecting task analysis data. The order in which the elements were arranged here is one that will maximize data gathering efficiency in most situations. Nevertheless, the best data gathering plan for any particular situation is the one that produces accurate task descriptions and minimizes the time that analysts and subject matter experts have to spend on the analysis.

4.2.5.4.1.4 Construct Data Gathering Instruments

To collect and record information supplied by subject matter experts, one needs a device that does two things: 1) provides a medium for recording the information, and 2) provides the subject matter expert with feedback on the state of the task descriptions. The first requirement is obvious. The second requirement comes from the fact that describing a task sequence to an analyst is something like thinking aloud to organize ideas into a meaningful and coherent train of thought. To describe a task, the subject matter expert has to have something analogous to what anyone normally does to get ideas on paper: that is, one makes rough notes, draws diagrams, or does both. One makes notes and diagrams for keeping the whole picture in mind. In this process the notes and diagrams provide a vehicle for keeping track of the developing train of thought, seeing where revisions are needed, and seeing where gaps need to be filled in. This is equivalent to the feedback that a subject matter expert needs for describing tasks in a task analysis.

In preparing to collect task analysis information, one must choose the best vehicle for recording information and simultaneously providing the subject matter expert with the type of feedback just described. The following are three common logical media for collecting initial task descriptions:

1 free-form notes describing the tasks;

2 a table format with columns representing task description items that can be filled in as the subject matter expert describes the tasks, and

3 any graphical format such as a block diagram or flow chart suitable for showing the sequence of tasks.

As the third medium implies, one or more of the function analysis techniques described in Section 27 (page 49) could be used to model a task sequence, and then serve as a basis for producing task descriptions.

The following are possible physical media for recording task descriptions and providing opinion to the subject matter expert:

1 a large format paper medium such as newsprint sheets that can be viewed by both the analyst and the subject matter expert;

2 a blackboard medium, and

3 a computer with a monitor display that can be viewed by more than one person, or a projection display.

The large format paper medium is probably the most flexible and easy to use. A blackboard medium is good for displaying and editing information, but the analyst must transcribe the information from the blackboard to a portable medium. Computer media have the advantage that the task information will be entered into the computer as the analysis is performed. The disadvantage of a computer is that it is less flexible than paper, or a blackboard, for reviewing the analysis information. The mechanics of operating the computer may also distract the analyst from attending to data collection demands.

Once the initial task descriptions have been collected, using computer-based media for collecting supplementary task information is usually becomes convenient. For example, task descriptions can be printed out in tables with blank spaces or columns that can be filled in by the subject matter expert, or by the analysts with help from the subject matter expert. Alternatively, the task description tabulations can be formatted and put on a computer display with input fields for entering supplementary task information.

Whatever the other merits of different data collection media, the chosen medium has to be something that the analyst can use without being distracted from concentrating on understanding the tasks being described by the subject matter expert.

4.2.5.4.1.5 Recruit Subject Matter Experts

A task analysis requires at least two subject matter experts, if only because relying on the views of just one expert is risky. From a statistical sampling perspective, the larger the sample of subject matter experts, the greater the generality of the information provided by the subject matter experts is. On the other hand, it takes time to get a subject matter expert accustomed to the analysis process. So having to break in more than a few experts may consume more time than can be justified by the increased generality of the task analysis information.

Another reason for keeping the number of subject matter experts small is that subject matter experts generally see functions and major tasks as coherent unified objects. When a subject matter expert breaks a task down into sub-tasks for the analyst, the subject matter expert has to start at the beginning of the task and work through to the task's completion. Picking up where another subject matter expert has left off part way through analysing a task is difficult for a new subject matter expert. Thus, it is best to recruit a few subject matter experts to work on complete functions.

The consideration of task unity also implies that subject matter experts should be recruited to work for at least a day at a time, and preferably for enough consecutive days to complete the analysis of one or more major functions. Breaking a subject matter expert's work into isolated sessions of one or a few hours will lose the continuity and momentum of the analysis; the same way as breaking up an analysis among several subject matter experts would.

An ideal subject matter expert is someone who has the following characteristics:

1 is a system user who has substantial recent firsthand experience at doing the tasks in the full range of the system's mission scenarios;

2 has experience in different operating units, and is therefore aware of differences in local operating procedures across the organization;

3 can grasp the nature of the task analysis process, and make the task analysis work as efficient as possible, and

4 does not have any biases that might lead to undue influence on the direction of the new system's design.

Since the project PM is responsible for obtaining subject matter experts, the analyst must make sure that the PM understands these requirements, and that the PM will try to recruit subject matter experts who can meet the requirements. Although recruiting subject matter experts who are ideal in every respect may not be possible, it is essential that they be current system users representative of the intended users of the new system.

Because the user organization's operations pressures usually make it difficult to free up personnel to participate in system analysis activities, recruiting subject matter experts should begin as early in the project as possible. The analyst should give the PM a two or three pages briefing memorandum explaining four things: 1) the rationale for doing the task analysis, 2) how the results will contribute to the system design, 3) the procedure for doing the analysis, and 4) the time that will be needed from the subject matter experts. This memorandum should be suitable for distribution to the user organization's commanders who will ultimately decide on providing personnel to act as subject matter experts.

When the subject matter experts have been identified, send them a briefing memorandum explaining the task analysis procedure. This memorandum can be the original briefing memorandum described in the last paragraph, or a suitably modified version of it.

Finally, even if the PM makes every effort to find subject matter experts that meet the recruiting criteria, the user organization's difficulties in providing subject matter experts may make it impossible to get consistently well qualified subject matter experts. Therefore, the analyst needs to be ready to work with whoever is available to act as subject matter expert.

4.2.5.4.1.6 Interview the Subject Matter Experts

Conducting the task analysis is a matter of executing the data gathering plan outlined in Section 4.2.5.4.1.3 (page 87). Although thorough planning will be the most important factor in having a smooth running task analysis, two factors remain difficult to deal with ahead of time. One concerns getting the analysis underway. The other is determining to what level of detail to analyse each task.

Even with the best of planning, getting started on collecting task data from subject matter experts usually involves some time-consuming trial and error. The trial-and-error period usually comes

from two sources: 1) getting subject matter experts accustomed to the task analysis process, and 2) refining the data collection instruments and procedures.

The most effective way to minimize time taken up with getting subject matter experts accustomed to the process is to give them the advance briefing memorandum described in Section 4.2.5.4.1.5 (page 91). Beginning the analysis by working through a function or task group that will likely be easy to analyse will help to get the subject matter expert used to the process. Generally, once a subject matter expert has worked through a function and has learned the process, the data collection should go as rapidly as the analyst's data recording process permits.

The procedures and instruments for recording the task analysis data will usually require some refining at the beginning of data collection. If the planned data collection procedure turns out to be altogether inadequate, a complete revision of the procedure will be needed. As just mentioned, once a subject matter expert becomes skilled at producing task descriptions, the data recording procedure will become the main limitation on the pace of work. Therefore, putting strong effort into making the data recording procedure efficient is well worthwhile.

It is a good practice to enter the task information from the subject matter experts into the data base storage medium as soon as possible after each data collection session. This will help to ensure that apparent errors or inconsistencies in the data can be checked with the subject matter experts while the data are still fresh. It will similarly save the analyst from having to depend on long term memory for task information that may not have been recorded on paper. Keeping the data base entries current with data collection will also ensure that any inconsistencies between the data collection procedure and database's structure are found and fixed early in data collection. If problems getting the data into the data base are found only after all the data have been collected, correcting the problems may be expensive or impossible.

How much detail into which to analyse tasks is often difficult to decide before doing the analysis, unless defining a rule for this decision has been possible (see Section 4.2.5.2, page 78). If no rule has been established in advance, one or more of the following checklist items could be used for deciding when a task has been analysed in sufficient detail. Thus, a task would be described in enough detail when the analyst can:

1 infer demands such as workload demands, cognitive demands, or physical demands;

2 derive quantitative objective measures of the adequacy of performance of the task;

3 identify potential ways of improving performance of the task;

4 identify personnel selection criteria or personnel training requirements for the task;

5 judge that a person, in principle, could use the task description to perform the task correctly.

In practice, analysts and subject matter experts typically have a good intuitive sense for when a task has been described in sufficient detail. Nevertheless, having systematic criteria for checking whether or not each function or task group has been analysed in enough detail is always useful.

Once all the task analysis data have been collected and entered into the data base, the analyst can go back to observe operations and verify whether the task analysis adequately reflects what system users do in operations. When this verification is completed, the task analysis will be complete.

4.2.5.4.1.7 Summary of Task Analysis Data Collection Procedures

Collecting data for a task analysis of an existing system consists of five steps:

1 Obtain and review all system documentation that can be of use in collecting and verifying task description information.

2 Create a data collection plan that includes provisions for:

- 2.1 reviewing documentation and identifying major functions and tasks;
- 2.2 observing operations to get an orientation to the operating environment;
- 2.3 conducting data gathering with subject matter experts to get initial narrative descriptions of the tasks;
- 2.4 verify task descriptions against documentation;
- 2.5 having independent subject matter experts verify the narrative task descriptions;
- 2.6 obtaining supplementary task information from subject matter experts, and
- 2.7 observing operations to verify the completed task analysis results.

3 Identify and construct the data gathering instruments that will be used to collect information from subject matter experts.

4 Recruit the subject matter experts, including:

- 4.1 arranging with the PM for getting subject matter experts from the user organization, and
- 4.2 preparing written briefings on the task analysis procedure for commanders and the subject matter experts.

5 Conduct the task analysis data gathering by executing the data gathering plan.

The task analysis procedures just described are for an existing system. In a CCIS development project, this analysis will be the baseline task analysis from which users' tasks in the new system will evolve. From this point on, descriptions of users' tasks will come from translating system design features into user performance requirements - in short, taking the system design and deducing descriptions of the tasks that the users will do to operate the system. The next section describes how to develop a task analysis from system design data.

4.2.5.4.2 Task Analysis Procedure for A System in Development

Keeping the task analysis up-to-date during system development is essential for conducting design reviews and constructing design tests. The task analysis procedure that will be described in the following presumes that the baseline task analysis has been completed, and that the results of the baseline task analysis have been used to guide the system design.

If for any reason design has got underway without a baseline task analysis, conduct a baseline analysis before undertaking any analysis of user tasks in the emerging system design. The reason that the baseline analysis must be the first step in the total task analysis effort is that user tasks directly associated with operating the CCIS tell only part of the story about how users perform their wider command and control tasks. Without a baseline task analysis context, descriptions of operating tasks in the new CCIS will be largely meaningless except for assessing the design against a few general HFE design principles. In particular, without a baseline analysis making significant HFE contributions to the design will be impossible, and deciding whether the new CCIS improves the performance of the overall command and control system will also be impossible. The baseline analysis also gives the HFE analyst the familiarity with existing operations that will be needed to anticipate the impacts that various system design features will have on operations.

The procedure for maintaining the task analysis during system development is conceptually the same as that for doing the baseline task analysis described in the preceding sections (beginning with Section 4.2.5, page 73). However, since the system design task analysis has to keep abreast of an evolving design, the analysis will be either a continuing process synchronized with the evolving design, or an iterative process synchronized with design phases.

The following outlines the steps in maintaining an up-to-date task analysis for a developing system.

1 Set up the system design task analysis database.

2 Identify sources of task information.

3 Acquire and review system documentation.

4 Plan the data collection procedure.

5 Construct data gathering instruments.

6 Collect the task analysis data.

The analysis represented by these steps can be repeated, in whole or in part, whenever the system design task analysis needs to be brought up to date. The next subsections describe each step of the analysis procedure.

4.2.5.4.2.1 Setting Up the System Design Task Analysis Database

Normally the baseline task analysis database structure can be used for the system design task analysis database. It will prove useful later to keep the system design task analysis database congruent with the baseline database. This will make it easy to compare tasks in the original command and control system with command and control tasks after installation of the new CCIS.

Keeping the system task analysis database congruent with the baseline database does not preclude adding task information items that might be useful for keeping track of the system design. For example, if a user interface design guideline is being used to standardize interface design features, referring to the interface design guideline in the task analysis database might be useful. Adding information items for the system design database will not destroy congruence between the information in the baseline database and corresponding information in the system design database. Removing information items or changing data field definitions will affect the congruence.

4.2.5.4.2.2 Identify Sources of System Design Task Information

Possible sources of information for the system design task analysis are:

1 system design documentation, and baseline mission, function and task analysis documentation,

2 functional prototypes or other representations of the design, and

3 system design personnel who can describe how the system design would work, or describe the operation of design prototypes.

The only system design documentation that can be relied on ahead of time for design task analysis data is the documentation specified in the project SOW. The documentation will consist of system analysis and design reports. These are roughly equivalent to the reports specified in the CCIS HFE Requirements DIDs. The degree to which this documentation will provide useful task analysis information depends entirely on the circumstances of the project. Right now, system engineering practice does not consistently demand documentation that can be depended on for task analysis data. Therefore, system documentation may not turn out to be a rich source of information for task analysis data. In fact, rather the reverse, it may turn out that the system development plan designates the design task analysis as the project documentation for how users will operate the system. This leaves three sources of information for producing the system design task analysis: 1) the baseline HFE analysis data, particularly the task analysis, 2) representations of the system design, and 3) system design personnel as the primary sources for task analysis data.

The baseline task analysis will not describe operating tasks associated with the new CCIS. Nevertheless, treating the design system analysis as a matter of modifying the task descriptions in the baseline analysis may be easier than developing an entirely new analysis. Using the baseline analysis as a foundation for the system design analysis also has the advantage of making it easier to identify baseline tasks that may be adversely affected by the impacts of the new CCIS.

Some form of system representation, preferably a functioning prototype, is a virtual necessity for a design task analysis. Narrative documentation describing how a system concept or design will work will not be detailed enough to provide accurate task descriptions at more than the broad function level. Even if the descriptions are fairly detailed, chances are high that what users do in the system portrayed in concept design documentation will change by the time a concrete design emerges. Instead of trying to develop task analysis data from such documentation, it is better to wait until the system design includes representations that are definitive enough to yield descriptions of users' tasks. This is not to suggest that the design task analysis should be left until a final system design emerges. Since the design task analysis is a primary medium for identifying design problems, it must be paced so that it can influence design. Thus, timing task analysis activities requires striking a balance between keeping the analysis in step with the design and having a design that is sufficiently definitive to yield accurate task analysis data.

A new CCIS will almost certainly be a computer-based system. Therefore, the concrete design will very likely include functional prototypes whose operating procedures will necessarily also represent the tasks that users will do with the new system. Design prototypes and their operating procedures provide a good basis for extracting user task descriptions. In fact, producing the task descriptions becomes a straightforward matter of documenting the design operating procedures in a task analysis format.

The most effective way to use a design representation in task analysis is to have the designer use the design representation to demonstrate the user's tasks, and then record the task descriptions. Normally systems are designed in functional modules with an individual designer, or a designated design team responsible for each module. Consequently the task analysis analyst can probably use mission, function and baseline task analysis information both to structure the task analysis, and to coordinate task analysis activities for each module with the individual or team responsible for designing the module. This amounts to identifying and employing design personnel to act as subject matter experts, in much the same way as users of an existing system serve as subject matter experts for a baseline task analysis.

4.2.5.4.2.3 Acquire and Review the System Design Documentation

As discussed in the previous section, system design documentation will not likely be a particularly useful source of documentation. Nevertheless, whatever documentation is available should be reviewed. In addition, the baseline HFE analyses should provide a good preview of tasks in the new system. As in an analysis of an existing system, review the documentation to identify major functions or task groups, and to create a preliminary task analysis database structure that identifies the major tasks and possibly some sub-task groups.

4.2.5.4.2.4 Plan the Design Task Analysis Data Collection Procedure

A data collection plan similar in principle to a data collection plan for an existing system (see Section 4.2.5.4.1.3, page 87) should serve for building up task descriptions for a system design. The following steps represent a data gathering plan for a task analysis of a system design.

1 Review documentation and identify major functions and tasks.

2 Identify and arrange for access to the design personnel who can provide access to design representations, and explain the operation of the representations.

3 Create a schedule for conducting data gathering sessions.

4 Work with design representations and design personnel to get narrative descriptions of the tasks, and to the degree possible, data on the supplementary task information items.

5 Verify the task descriptions with design personnel, and against documentation.

As discussed earlier, during the early stages of design there will not likely be much documentation for use in the task analysis.

It is important to arrange early in the project for contacts with individual design personnel who can act as design subject matter experts able to demonstrate and explain the operation of different system modules. It is also important to ensure that the availability of design personnel is part of the system development plan, and that time is allocated for them to participate in keeping the task analysis up to date. Finally, design personnel should be provided with a briefing memorandum similar to the one for briefing subject matter experts in the baseline task analysis (see Section 4.2.5.4.1.5, page 91).

As already pointed out in Section 4.2.5.4.2.2 (page 96), timing of data gathering sessions is critical because the system design may be continuously evolving. On the one hand, data gathering sessions have to be scheduled so that the analysis for each system module is done when the design of the module is complete or nearly complete for the phase of the development. On the other hand, data gathering must but not be left until it is too late to fix any design problems identified from the analysis. If modules are being developed sequentially, the task analysis work can be scheduled so that each module is analysed as it is completed. If modules are being designed in parallel so that all modules will be completed just before the end of a design phase, then the HFE analyst may not have time to complete the analysis before the end of the phase. If so, it may be necessary to build up the task analysis of each module as the module is being designed. This approach will require extra close liaison with design personnel to make sure that task analysis sessions are as productive as possible.

Gathering task information using a design representation as a data source medium can be done by using either of two techniques:

1 the designer operates the representation to simulate tasks, and the analyst records the tasks based on the designer's explanations or commentary, or

2 the designer shows the analyst how to operate the representation so that the analyst can then independently perform operating tasks and record the tasks at leisure.

The first approach is more demanding in total labour, but it guarantees an accurate analysis. The second approach is less labour demanding, but it makes the analyst vulnerable to errors in recording the tasks. To make the second approach work, the analyst has to make sure he or she understands how the representation works before starting to record the task information. In

practice, it is likely that an analyst will use both techniques; choosing one or the other depending on the nature of the design module being analysed.

As a matter of good practice, the system design task descriptions should be verified with the contributing design personnel when the descriptions have been drafted.

4.2.5.4.2.5 Construct Design Task Analysis Data Gathering Instruments

Any of the instruments used for data collection in the baseline task analysis can be used for the design task analysis (see Section 4.2.5.4.1.4, page 90), that is, free-form notes, a table, or any graphical format. However, in a design task analysis, there will likely be more freedom to use a recording medium that allows putting data directly into the task analysis database. This is especially so if the analyst has independent access to the design representation and can record task information without taking up the design subject matter expert's time.

4.2.5.4.2.6 Collect the System Design Task Analysis Data

Like conducting the baseline task analysis, conducting the system design task analysis is a matter of executing the data gathering plan. However, several factors should make the system design task analysis easier to do than the baseline task analysis. First, the analyst will already be familiar with the command and control system from the baseline analysis work, and therefore, will not have to struggle with gaining familiarity with the system being analysed. Having the baseline task analysis in hand will eliminate having to experiment with data formats and recording media. In addition, the task hierarchy established in baseline task analysis, particularly the requirements for levels of task description detail, will provide a ready-made basis for a task description hierarchy in the system design task analysis. All of these factors will help to reduce the time consumed to get the data collection started.

Another factor that can increase efficiency is having access to design representations in a way that permits recording task information without the designer having to be continuously present. With this freedom, it becomes feasible for the analyst to enter task information directly into the task analysis database: for example on a portable computer. This situation eliminates the need to have a recording medium that can be displayed to the subject matter expert, and it eliminates transcribing data from the data collection instrument into the database.

A factor that can make a system design task analysis more difficult than a baseline analysis is that the state of a system design is continually changing. In contrast, the state of a baseline system remains static. This makes the scheduling of data gathering for a system design task analysis more critical than it is in a baseline analysis. Although design personnel may not be in principle any harder to get access to than user organization subject matter experts, the constraints of timing may still make it harder to get time from design personnel. Consequently, the analyst must be sure to keep close track of the system design progress and make sure that design personnel and system representation will be available when needed.

Once all the task analysis data have been collected, entered in the data base, and verified, the system design task analysis for the design phase will be complete.

4.2.5.4.2.7 Summary of System Design Task Analysis Data Gathering

Performing data collection for a task analysis of a system design consists of six steps:

1 Define the task analysis database structure based on the one used for the baseline task analysis.

2 Identify sources of task description information including documentation, design personnel who can describe the design, and design representations that can serve as media for extracting task information.

3 Acquire and review the system design documentation that can contribute to the task analysis.

4 Create a data collection plan that includes provisions for:

4.1 reviewing documentation to identify major functions and tasks;

4.2 identifying and arranging access to the design personnel and design representations;

scheduling times for conducting data gathering sessions;

4.3 procedures for working with design representations and design personnel to get task descriptions, and

4.4 verifying the task descriptions with design personnel, and against documentation.

5 Identify and construct the data gathering instruments that will be used to collect task description data.

6 Conduct the task analysis data gathering by executing the data gathering plan.

4.2.5.5 Use of Other Task Analysis Methods

So far in this section (Section 4.2.5) task analysis has been treated from the perspective of the critical-task type of task analysis specified in the CCIS HFE Requirements. While the CCIS HFE Requirements specifies a critical-task type of analysis, the CCIS HFE Requirements does not preclude applying any additional type analysis that would be helpful in understanding the tasks that users do in a system. Since there is no universal HFE terminology limiting task analysis to specific types of analyses (see discussion at the beginning of Section 4.2, page 31), there is also no reason to limit the analysis to that specified in the CCIS HFE Requirements.

Other methods that can be used to analyse tasks include all the function analysis methods described in Section 4.2.4 (page 49). Although function analysis methods are described here as methods for analysing the major functions of a system, they can also be used to analyse user tasks. Using any of the function analysis methods to analyse tasks is only a matter of applying the

analysis to user tasks, or doing the analysis in sufficient detail to isolate user task performance from other system functions.

The most likely time when a function analysis method will be useful is when a method suggests itself as the best way to study a particular task performance problem. For example, command and control planning tasks suggest applying one of the flow process function analysis methods. Monitoring tasks involving information flows among headquarters personnel suggest a network analysis method. Conversely, it was pointed out in Section 4.2.5.4.1.4 (page 90) that function analysis techniques can be used to draw out task descriptions from subject matter experts. Thus, a function analysis method can be an instrument for collecting task data. However, it is unlikely that any one function analysis technique would serve as the best medium for analysing all a user's tasks.

On the whole, the task analysis approach specified in the CCIS HFE Requirements provides a sound general basis for analysing all the tasks that different personnel do in command and control operations. Other types of analyses such as function analysis methods can be applied whenever they are needed to amplify or clarify user behaviour.

4.2.5.6 Summary of Task Analysis

The following summarises how to do task analysis activities to conform with the CCIS HFE Requirements.

1 The user task analysis provides a continuing description of the tasks that command and control personnel perform in operations through from the baseline predecessor system to implementation of the new CCIS. As such, the task analysis provides the basis for all HFE design and design testing activities throughout the CCIS development project.

2 The CCIS HFE Requirements (5.1.1.4) specifies a type of hierarchical task analysis usually called a critical task analysis. The analysis specifies six items of information describing each task, and a further 10 items describing related groups of tasks. This provides what amounts to an inventory of information about users' tasks. The task information is intended for use in HFE activities such as: contributing to requirements analysis and function allocation, defining measures of user performance and system effectiveness, and evaluating alternatives for CCIS design.

3 The CCIS HFE Requirements does not define what makes up a task, nor how to show relationships among tasks. Although the HFE literature contains a variety of definitions, there is a consensus that a task is essentially a purposeful transaction between a human operator, or user, and a system. This, with the addition of a provision for expressing sequential relationships among tasks, provides a conceptual basis for setting up a task analysis database.

4 Setting up a task analysis database consists of defining a logical structure for the database, and a physical medium for storing and displaying the task analysis data. A database logical structure suitable for a CCIS HFE Requirements task analysis contains a record for each task,

and record fields containing the items of task description information, along with fields containing task sequence and hierarchy data. Physical media suitable for task analysis data include manual spreadsheets, computer spreadsheets, database programs, and custom software.

5 During system development, task analysis data are collected at two stages: 1) at the beginning of development to create a baseline task analysis, and 2) during development to track users' tasks in the emerging system design. The procedure for collecting baseline task analysis data contains six activities:

5.1 identifying sources of task information;

5.2 acquiring and reviewing relevant system documentation;

5.3 constructing a data collection plan that covers: 1) identifying major functions and tasks, 2) observing operations, 3) work with subject matter experts to get initial narrative task descriptions, 4) verify task descriptions against documentation, 5) verifying task descriptions with independent subject matter experts, 6) work with subject matter experts to complete supplementary task description items, 7) verifying the completed analysis with subject matter experts, and 8) verifying the analysis by observing operations;

5.4 constructing data collection instruments;

5.5 recruiting subject matter experts, and

5.6 executing the data collection plan.

6 The procedure for collecting task analysis data for a system under development contains six activities:

6.1 setting up the task analysis database;

6.2 identifying sources of task description information including: 1) documentation, 2) design prototypes, and design personnel;

6.3 acquiring and reviewing design documentation;

6.4 constructing a data collection plan that covers: 1) identifying major functions and tasks, 2) getting access to design personnel and design representations, 3) scheduling data gathering sessions, 4) work with design representations and design personnel to get task descriptions, and 5) verifying the task descriptions;

6.5 constructing data collection instruments, and

6.6 conducting the task analysis by executing the data collection plan.

Although the CCIS HFE Requirements specifies a critical-task type of task analysis, it does not preclude the use of other methods such as function analysis for studying user behaviour. Other analysis methods can be used to study user functions or task groups for which the methods are particularly well suited, or to explore system design problems associated with user performance, or to serve as media for collecting task descriptions.

A task analysis describes the individual tasks that users do, and to a limited extent, describes the gross sequential and timing properties of task performance. However, a task analysis cannot describe the dynamic characteristics of user performance either in existing operations or in anticipated operations. The next section describes methods for analysing the dynamic characteristics of task performance.

4.2.6 Analysis by Computer Simulation

This section describes methods for analysing the dynamics of users' tasks, particularly tasks driven by periodically or randomly occurring events. Some of the analyses described in this section will be ones called task analysis in Beevis *et al.* [18] and in Luczak [32], and called function analysis in Laughery and Laughery [25]. Elsewhere, Chub *et al.* [47] and Laughery and Corker [48] call the analyses in this section computer modelling of task performance.

Describing the dynamics of user tasks is important because task dynamics determine users' workload capacities: that is, how quickly users can perform not only individual tasks, but also how quickly they can perform complex sequences and combinations of tasks. The operational sequence analyses in this section are analyses that can be used to study the workload implications of users' tasks. From a system design perspective, workload demand depends on two independent factors: 1) the system design, and 2) the rate at which system inputs demand action from the user.

System design affects user workload capacity in two ways: 1) by the HFE quality of the design, and 2) by the match between system capacity and workload demand. Good HFE design reduces the work needed to do tasks and accomplish task goals. Therefore, good HFE design maximizes workload capacity. Irrespective of HFE design quality however, the system also has to have enough personnel and facilities to handle the maximum volumes of workload demand that are likely to be met in operations. To ensure that the system has enough capacity to deal with demands, designers need to know two things: 1) the volumes of demand that the system will eventually have to meet, and 2) the system's unit workload capacity, including the workload capacities of the users who handle the work.

Designing a system to increase workload capacity and reduce workload demand raises several HFE design questions. Which are the user tasks that have the most impact on overall system workload capacity? How could changing these tasks affect workload capacity? How would different design options affect user workload capacity? What is the relationship between operations environment variables and user workload demand? What would be the best way to organize user tasks to maximize user workload capacity?

These types of questions can be answered in two ways. One is by doing empirical experiments. The other is to create a model that can simulate system and user behaviour, and then see how the model behaves in operations scenarios. Although empirical experiments are usually preferable, they are also expensive - and often impractical. Simulation models on the other hand, are comparatively cheap. Once a simulation model has been constructed, it can generate data for

almost limitless numbers of iterations and variations of both operations and system configuration scenarios. The remainder of this section describes simulation models that are useful for studying workload problems, and deals with the following topics:

1 the CCIS HFE Requirements specifications for using simulation methods;

2 descriptions of simulation methods applicable to HFE analysis, including:

  2.1 task network simulation methods and models,

  2.2 operations management models,

  2.3 purpose-built simulations;

3 the use of operations statistics to estimate values of simulation model parameters, and establishing the reliability and validity of simulation models.

4.2.6.1 CCIS HFE Requirements Specifications for Computer Simulation Analysis

The CCIS HFE Requirements (5.15.c) specifies the use of simulation methods as a way to verify decision support requirements or function allocations that cannot be verified or supported from HFE literature. However, as circumstances warrant, simulation methods can also be used to extend any of the mission, function and task analyses specified in CCIS HFE Requirements (5.1.1). To the degree that a simulation can serve the purposes of a system design prototype, a simulation model is applicable wherever the CCIS HFE Requirements specifies the use of prototypes, particularly in CCIS HFE Requirements (5.2). Finally, since simulations commonly provide data on workload demands and workload capacity, simulation models are applicable wherever the CCIS HFE Requirements specifies measures of user workloads: CCIS HFE Requirements (5.1.4.1.2, 5.1.4.2.1, 5.1.4.2.2, 5.2.2.2, 5.2.4.g, and 5.2.5.c to e).

The CCIS HFE Requirements has one DID that specifically concerns the use of simulation methods. This is the DID titled Human Factors Engineering Simulation Plan. This DID specifies how to report plans for using prototypes, engineering simulators and mockups in human factors engineering analysis, design, and testing activities. Plans for using any of the simulation methods described in this CCIS HFE Design Guidelines should be reported according to the specifications in the Human Factors Engineering Simulation Plan. The results of simulation-based analyses can be reported according to whatever DID covers the purpose of the analysis.

4.2.6.2 System Analysis of Integrated Networks of Tasks (SAINT)

SAINT is described here as one example of the many computer-based simulation methods available for analysing human performance in operational sequence analysis.

SAINT is a method for simulating the performance of any task sequence that can be represented as a task network. A task network representing dialling a telephone is probably the most common illustration of a task network suitable for analysis by SAINT. Figure 19 shows an example.

The literature examples of SAINT task networks for dialling a telephone, as well as illustrating a task network, will also show that there are many different ways of breaking down the apparently simple task of dialling a telephone. Some examples are much more detailed than the example in Figure 19. Others are a little simpler. Taken together, the examples show that there are usually many different ways of representing the same operating task.

The SAINT method has been incorporated into a computer program that can, in effect, activate each network task in a sequence according to probability rules defined by the program user. In Figure 19, for example, one could specify that the probability of incorrectly dialling a number is 0.1. The program would then, on average, trace the 'pick up phone', 'dial number', 'hang up', and back to 'pick up phone' cycle once every 10 calls. The SAINT program user can also specify how long it takes to do each task. Then, by repeating the dialling process, the user could accumulate statistics on the times required to do the task as a whole, and the times taken to do different chains of sub-tasks. By altering the time requirements for individual tasks, or altering the probabilities of task outcomes, the program user could explore different task form scenarios. When the altered task parameters represent anticipated system design impacts, the simulation becomes a way of exploring system design options.

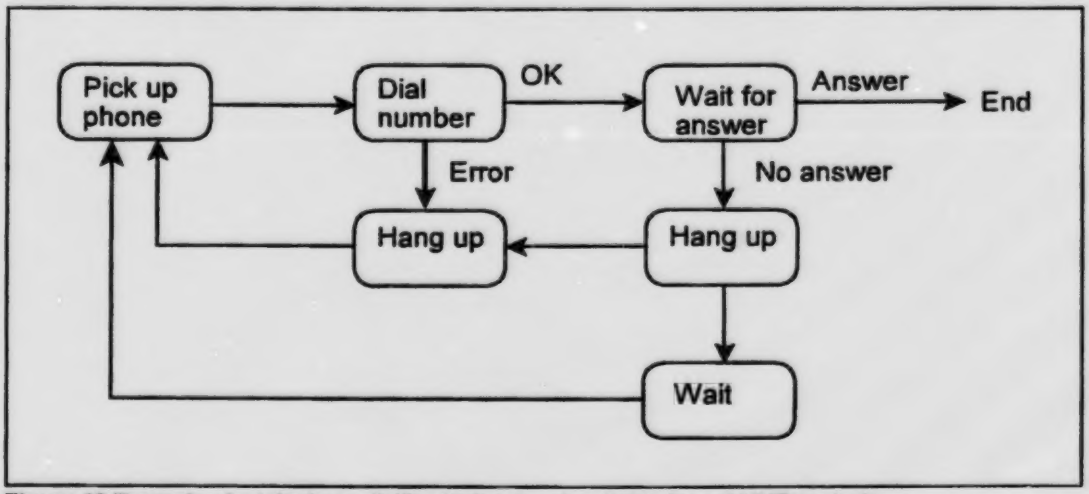

Figure 19 Example of a telephone dialling task network suitable for a SAINT analysis.

SAINT was originally set up as a mainframe Fortran program. It is now available as a microcomputer program called *Micro-Saint*. The operation of the program is described in Laughery and Corker [48]. To set up a simulation, the program user first constructs the task network (like the one in Figure 19) defining the sequential relationships among the tasks. The program user then defines four properties for each task.

1 *Time to perform the task.* This can be a fixed time or a distribution of times that the program samples each time the task is activated in a simulation run.

2 *Release condition.* This is an optional property that prevents the task from starting unless some system state is true: for example, the operator being available to perform the task.

3 *Beginning effect.* This is another optional property that can set a system state when the task begins: for example, make the operator unavailable for performing any other task.

4 *Ending effect.* Similar to the beginning effect, this property sets a system state when the task ends: for example, makes the operator available to perform other tasks.

Besides these properties, SAINT defines several housekeeping properties such as the task name and number, and specifications for displaying task execution.

Tasks that branch to more than one subsequent task have an additional property. For these tasks, the program user can specify one of three types of branching conditions.

1 *Probabilistic.* The branch path is selected according to a probability distribution specified by the program user.

2 *Tactical.* The task will branch to the task having the highest value according to a choice decision rule specified by the program user.

3 *Multiple.* The task will simultaneously branch to all the subsequent tasks to which the branching task is linked.

Once the task properties and various parameters have been assigned, the program user can run the simulation by triggering the network's start task. SAINT has debugging facilities that allow the user to check that the task network is functioning properly. Once debugged, the user can then run repeated SAINT simulations and collect the various performance statistics that the program generates from the runs.

Statistics produced from SAINT include times to complete tasks, resource utilization statistics, histograms of task completion times and intervals between task activations, and other statistics. All the data produced from a SAINT simulation are time data representing things like how long it takes to perform task sequences, and the percentages of time taken up by the performance of different tasks. Where a percentage of time (i.e., resource utilization) statistic represents the percentage of time that a system user is busy doing a task, the statistic can be treated as a measure of the workload imposed by the task.

Laughery and Corker give an example of *Micro Saint* applied to a process control monitoring task. Other examples of SAINT applications can be found in McMillan, *et al.* [49]. Laughery and Corker also give examples of extending SAINT to more direct measures of workload variables. One example is a simulation developed by Drews *et al.*[18] to predict the mental workloads that would be imposed on the pilot of an attack helicopter that was being considered for development. The purpose of the simulation was to help determine whether pilot workload demands would

[18] The reference given by Laughery and Corker [48] for this work is: Drews, C., Laughery, R. R., Kramme, K., & Archer, R. (1985). LHX Cockpits: Micro SAINT simulation study and results. Report prepared for Texas Instruments - Equipment Group, Dallas, Texas.

allow the helicopter to be operated by a one-person crew instead of the two-person crew that is usual in an attack helicopter.

The simulation was constructed by incorporating a theoretical model of mental workload into the SAINT model. The theoretical model was Wickens' multiple resource theory [50] (see also Tsang and Wilson [51], and Wickens and Carswell [52]) which asserts that mental workload is a weighted sum of the workloads from four mental processes: 1) visual processes, 2) auditory processes, 3) cognitive processes, and 4) psychomotor processes. This model was incorporated into the SAINTS' simulation by quantifying the workloads of different helicopter operating tasks using a scaling system developed by McCracken and Aldrich[19].

McCracken and Aldrich's scaling system assigns rating values to tasks depending on their attentional demands. For example, a simple visual task like visual monitoring has a value of 1. A complex visual task like reading text has a value of 7. Auditory, cognitive, and psychomotor tasks are similarly rated according to attentional demand. The attentional demand of any task is then the sum of the ratings for the task's visual, auditory, cognitive, and psychomotor components. In turn, the attentional demand imposed on an operator at any instant is the total of the attentional demand ratings for all the tasks the operator is performing at that instant. During a simulation run, the attentional demands of a task are detected through the task's Task Beginning Effect and Task Ending Effect. When a task starts, its attentional demands are added to the accumulator variables representing the operator's total visual, auditory, cognitive, and psychomotor attentional demands. When the task ends, its attentional demands are subtracted from the accumulator variables.

The simulation results suggested that a substantial proportion of operations would generate workloads higher than a one-person crew could probably deal with comfortably. This finding agreed with findings from other research.

The helicopter example shows that the task network concept embodied in *Micro Saint* can support a wide range of design problems. For example, Laughery and Corker describe the extension of helicopter simulation approach to the US Army's HARDMAN III engineering tool set for developing crew-operated systems. In this tool set, task network software is used to predict three quantities: 1) workloads of crew members, 2) performance time requirements, and 3) requirements for access to controls and displays.

In summary, SAINT is a well-established method for modelling task networks; and *Micro Saint* provides a ready-made medium for doing SAINT simulations. *Micro Saint* seems easy to use compared with computer-based simulation packages of not many years ago. However, recent developments in high level computer languages also make it relatively easy to construct purpose-built simulations (see Section 4.2.6.4, page 114 in the following), which may sometimes be easier to do than learning and adapting a commercial package.

---

[19] The reference given by Laughery and Corker [48] for this work is: McCracken, J, H., & Aldrich, T. B. (1984). Analysis of Selected LHX Mission Functions: Implications for Operator Workload and System Automation Goals. Technical Note ASI 479-024-84(b) prepared by Anacapa Sciences, Inc., June 1984.

SAINT is only one example of many general purpose programs available for doing simulations. New programs appear almost daily. When considering a simulation, it is always worth searching databases to see if there is software that closely matches the requirements of the simulation problem.

4.2.6.3 Service System Models

The discipline of Operations Management contains many methods and models that can be useful in analysing command and control systems [53]. This section describes the application of two main types of models:

1 critical path models for dealing with command and control planning activities, and

2 service system models for dealing with command and control monitoring activities.

4.2.6.3.1 Critical Path Analysis Methods

Critical path or PERT methods are among these operations management methods. The critical path chart was introduced earlier in Section 4.2.3.4 (page 36) as a mission analysis technique. Cameron [54] has illustrated the use of the critical path method, particularly the use of precedence diagrams, for studying command and control systems. A critical path chart, and its associated output variables, is the output from computations done on a task network not unlike the task networks that are the subjects of a SAINT analysis. In a critical path network, activities or tasks are specified according to their completion times, and precedence rules that specify for a given task which other tasks must be completed before the given task can begin. In the earlier mission analysis example of a command and control planning process, the flow process chart of the planning process (see Figure 2, page 37) defined the task precedencies. The basic critical path computations generated the critical path chart of the planning process shown in Figure 3 (page 39).

The critical path chart in Figure 3 was generated by a simple computer program for doing critical path computations. The input to the program is a data list specifying for each task: 1) the task's name, 2) its completion time, and 3) the names of the task's immediate predecessor tasks. As such, the program is a vehicle for simulating, in this instance, a command and control planning process. Thus, the program could be used to explore the effects of varying the arrangement of tasks in the planning process, or varying the completion times of tasks.

Beyond simple critical path analysis, operations management literature contains a wide range of techniques for modelling project-oriented activities like those contained in the planning phases of command and control operations. These other models deal with such things as personnel and resource requirements; personnel and resource utilization, optimizing resource demands over the period of a project, and like problems found in project management activities.

4.2.6.3.2 Service System Models

Service system models, which are also called facility capacity models, are another set of operations management models that can be useful in command and control system analysis. Two types of models will be described in the following: 1) analytic models, and 2) purpose-built simulations of service systems. Analytic models cover a variety of basic service system situations for which there are analytic functions whose solutions can describe the system's behaviour. Analytic models are essentially a set of mathematical functions created to solve service system quantities for a variety of standard systems. Thus, they are not strictly simulation models in the sense that Monte Carlo runs have to be used to estimate the values of the model's behaviour variables. However, most analytic models require a computer to do the calculations unless one is prepared to do what would be now considered an inordinate amount of work to solve the equations by hand. Therefore, the difference between running an analytic model and running a Monte Carlo simulation model is usually insignificant.

When the service system is too complex or unusual to fit a standard analytic model, a purpose-built model of the system can be created to simulate the system's behaviour. A SAINT task network model can be treated as a service system model. The following sections describe each type of service system model, and how it can be used in command and control system applications.

4.2.6.3.2.1 Analytic Models of Service Systems

A grocery store checkout counter is a familiar example of a service system. Seven basic variables describe the management of service systems. Of these, four are dependent variables: 1) how long a customer has to wait to get served, 2) how many customers are waiting to get served, 3) how long it takes to serve a customer, and 4) the percentage utilization of the server. Three are independent variables: 1) the rate of customer arrivals, 2) the time it takes to serve a customer, and 3) the number of servers.

Command and control operations contain a variety of examples of this type of service system. One example is the headquarters message handling system described in Section 4.2.3.6.2 (page 41), and illustrated in the flow chart in Figure 4 (page 44). In the message handling process, messages coming through the headquarters are equivalent to service system customers. Duty officers who handle the messages are equivalent to servers.

Applications of service system models use queuing theory mathematics to calculate relationships among a service system's dependent and independent variables. The following are some examples of relationships that are of interest in the headquarters message handling process:

1 the average length of time a message will wait in the message queue until it is processed as a function of time it takes a duty officer to process the message;

2 the percent utilization of duty officer time as a function of the time it takes to process a message;

3 the average overall length of time it will take to get a message processed as a function of the rate of message arrivals, and

4 the maximum rate of messages that can be processed without increasing duty officer utilization above a predefined percentage point, and without causing any message to have to wait more than a predefined time to be processed.

These four examples, along with others that are easy to imagine, represent relationships that are clearly important in understanding the capacity of a headquarters message handling system. These types of relationships are similarly important for designing a headquarters system so that it will have the capacity to handle anticipated message traffic.

The relationships among service system variables range from simple linear (including monotonic curvilinear) relationships that can be calculated from analytic functions, to complex nonlinear functions that can only be estimated from computer simulations. Figure 20 shows two examples of linear relationships for a message processing example.

The example in Figure 20 shows average wait times to process a message (on the lefthand Y axis), and percent duty officer utilization (on the right-hand Y axis), as a function of message processing time (on the X axis). In this example, wait times increase roughly exponentially from almost zero to 25 minutes as the time required to process a message increases from one minute to five minutes. Percent utilization on the other hand, increases arithmetically as a function of processing time. The example was calculated for messages arriving at random intervals at an average arrival rate of 10 messages per hour. Thus, Figure 20 represents one member of a family of graphs that could be generated for different average message arrival rates.

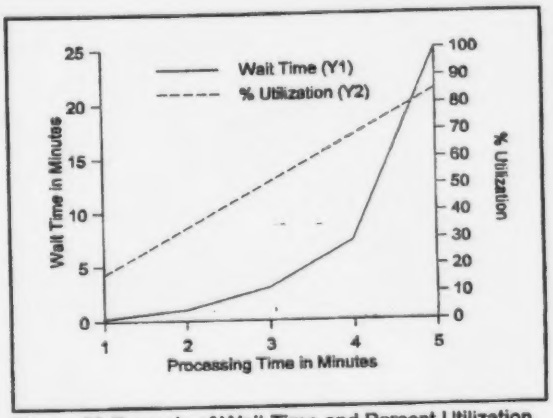

Figure 20 Example of Wait Time and Percent Utilization as a function of Processing Time.

In command and control system terms, utilization represents duty officer workload. Figure 20 shows that there can be sharp tradeoffs among workload, wait time, and message processing time. For instance, decreasing the message processing time requirement from five minutes to four minutes produces a drastic reduction in wait time, but only a proportionate reduction in the workload. Looking at the graph from another angle, keeping wait times low can be achieved only by keeping the duty officer's workload well below the officer's workload capacity.

The relationships illustrated in Figure 20 help to explain some results from a trial of an automated headquarters CCIS [55]. The CCIS being tested was installed in a battalion level command post. One measure in the trial was how long it took information in a message originating from a

battalion subunit to get through the battalion command post to the brigade headquarters[20]. Figure 21 contains a histogram showing the distribution of the times taken for messages to pass from subunits through battalion to brigade.

The histogram shows transmission times ranging from a few minutes to more than 45 minutes. Almost 20 percent of the messages took 45 minutes or longer. This even if the average time required to process a message in the battalion command post was about 2.5 minutes. The long transmission times came from the fact that messages arrival rates were close to the limit of the duty officer's workload capacity. This in turn caused the build-up of long wait times in the message queue. Service system calculations showed that message processing time would have to be almost halved to get transmission times down to a range where most messages would get through in less than about 10 minutes. Halving processing times appeared to be an impractical goal.

As a matter of incidental interest in connection with the battalion command post, the service system calculations showed that the simplest way to reduce transmission time would be to have a command post system that would let both duty officers work in parallel instead of in the traditional series arrangement (see Figure 4, page 44). In service system terms, this would be equivalent to having two parallel servers instead of two servers working in series.

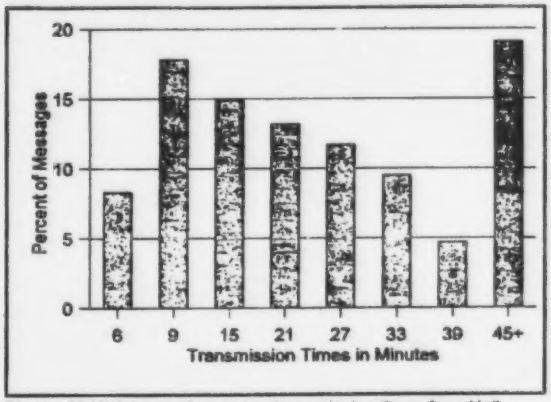

Figure 21 Histogram of message transmission times from Unit through Battalion to Brigade during a command post exercise.

4.2.6.3.2.2 Purpose-built Simulations of Service Systems

The service system calculations used for these examples were done with analytic functions that can be found in standard operations management texts (e.g., [53]). Analytic solutions are available for many simple service system configurations such as a set of checkout counters, fixed chains of processing operations, and the like. When the system to be analysed does not fit one of these standard configurations, the modeller must construct a Monte Carlo simulation that can produce estimates of the system's performance statistics. The following is an example of this approach.

The modelling problem was to estimate the workloads of a marine traffic regulator whose function was to monitor marine traffic and give vessels navigation directions. The regulator used a radar display and monitored communications traffic to keep track of vessels in the regulator's assigned marine sector. Largely, the regulator did much the same tasks as an air traffic controller does.

[20] A brigade is one level above battalion in the command and control hierarchy.

The model was constructed to represent four main sets of variables: 1) the physical configuration of the sector, 2) vessel traffic patterns, 3) times required for the regulator to do various tasks such as issuing clearances, and 4) variables representing the movements of vessel traffic. The model was constructed to generate vessel traffic, and then based on vessel positions calculated at regular clock intervals, generated demands for service from the regulator. For example, a vessel passing a designated call-in point would generate a demand for a clearance, or two vessels meeting or passing would generate a demand to monitor communications between the two vessels. Once traffic was set in motion, the model collected data on the tasks performed by the regulator. These data provided estimates of the regulator's workloads under different traffic conditions, and different sector configurations.

Figure 22 contains a graph illustrating data collected from the simulation. This simulation, which was for a particular sector configuration, had two independent variables: 1) the average time needed for the regulator to respond to service demands such as clearances, and 2) the volume of vessel traffic. Figure 22 Shows four graphs: one for each of four average service times. The X-axis of the graph shows the traffic volumes scaled in two ways: 1) number of vessels entering the sector per hour, and number of vessels that would be in the sector at any one time.

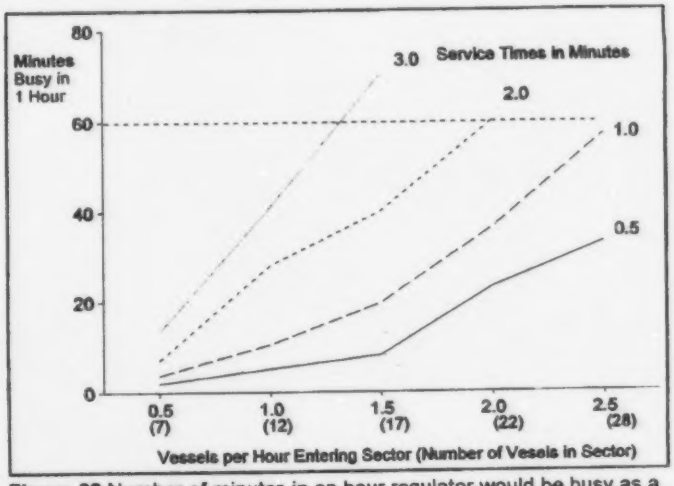

Figure 22 Number of minutes in an hour regulator would be busy as a function of rate of vessels entering sector and times required to service vessels.

The Y-axis in Figure 22 shows the average number of minutes out of the hour that the regulator would be busy servicing vessels. Each individual point in each graph represents a simulated 60 hours of traffic. The 60-minute cutoff line on the graph shows the point at which the regulator would begin building up a backlog of service demands that the regulator would never be able to clear. In fact, the cutoff would need to be much lower than 60 minutes because the simulation did not include one of the regulator's most important tasks: monitoring the radar.

A primary purpose of the simulation was to see how much time would be available for the regulator to do monitoring tasks under different traffic conditions. This simulation approach was taken for a combination of three reasons: 1) the regulator was expected to monitor traffic whenever time was available, 2) the regulator had no standard operating procedures defining how to do monitoring tasks, and 3) there was no standard for how much time should be devoted to monitoring tasks. In short, the operations management system had no criterion for deciding when the traffic volume would be greater than a regulator should attempt to handle. As it later turned out, the point on the one minute service time graph corresponding to 22 vessels in the sector

represented the traffic volume that most regulators could handle without losing comprehension of the general traffic situation.

Although the marine traffic regulation is not exactly a command and control operation, translating the simulation into a command and control environment is not difficult. For example, the variables in the marine model (the physical configuration of the sector, vessel traffic patterns, times required for the regulator to perform various service tasks, and variables representing the movements of vessel traffic) have parallels with a headquarters' monitoring operation. A duty officer processing messages and keeping track of operations on a battle map corresponds to the regulator handling vessel service demands and monitoring traffic on radar. Conceptually, battlefield configurations and sectors are not particularly different from marine traffic sectors. Tactical manoeuvres and not conceptually different from vessel traffic movements. Finally, one of the duty officer's most important tasks, maintaining comprehension of the battle situation, is something typically done with time available when message traffic permits - just as the marine traffic regulator monitored traffic when servicing traffic permitted. Indeed, the command post trial discussed earlier [55] showed that duty officers' comprehension of the battle situation suffered considerably when message handling demands reached levels that were still well below the time nominally available for handling messages.

Something that makes command and control simulations particularly feasible is the fact that command and control exercises are primary means of training command and control personnel. These exercises are in fact simulations in which a headquarters staff do the tasks of planning and monitoring an operation based on scenario information fed into the headquarters by a team of exercise controllers. Scenarios are prepared ahead of time as scripts that the controllers use to create messages to send to a headquarters. These scenarios, along with records created during exercises, provide a good source of information for constructing more abstract simulations of command and control operations.

4.2.6.3.3 Summary of Operations Management Models

Operations Management literature provides techniques that can be useful for analysing command and control operations. Two sets of operations management techniques are useful in analysing command and control operations.

1 *Critical path methods* are useful for analysing the project management features of command and control, particularly planning activities.

2 *Service system analysis* methods are useful for analysing activities in the monitoring phase of command and control operations. Service system analyses provide models from which measures of, among other things, a system's workload capacity can be defined. Two types of models are available:

   2.1 analytic models that fit many common situations, and include mathematical functions that support calculations of such things as workload capacity as a function of system configuration, and

2.2 purpose-built simulations that model system configurations and generate system performance data as a function of simulated service demands.

Understanding and being able to measure command and control personnel workloads are fundamental to the HFE contribution to system design. Operations management methods provide well-established techniques for analysing the workload features of both existing and proposed systems.

4.2.6.4 Simulation by Purpose-built Software

Writing a custom simulation program is necessary when none of available simulation software can handle the simulation problem. A custom program may also be the best option when writing a program would be easier than acquiring and learning to use existing simulation software.

The first situation may become obvious when the simulation problem is compared with available commercial software. If the problem is clearly different from the problems that any available simulation program is designed to handle, writing a custom simulation program is the only option. In other cases, the adequacy of available simulation software may be uncertain. One example is where limitations in the functionality of the commercial software make it impossible to simulate the system sufficiently accurately. For example, if the statistical distribution of a variable in the simulation problem can only be approximated in the commercial program, one is left with a question about what effect the program's the lack of distribution fidelity will have on the generality of the simulation results. Answering this question may require much statistical homework, or experimentation to validate the program's results against real system performance. If so, the effort to write a custom simulation that accurately simulates the variable's distribution might turn out to be less than the effort needed to work around the commercial program's limitations.

Writing a custom simulation can also be preferable when the simulation problem is relatively simple and the available simulation software has much more functionality than needed. Writing a program might also be preferable if the ready-made software would also take substantial effort to obtain, learn and set up.

The following outlines the architecture of a simulation program. A simulation program needs four code modules and a documentation module:

1 an input module for setting the values of initializing parameters, and for starting the simulation runs;

2 a runtime display to show the progress of the simulation run;

3 a program module containing the functions and routines that simulate the system behaviour;

4 an output module that displays results online and saves results in an output file, and

5 sufficient documentation to describe how to run the program, and to permit modifying or reusing the program code when needed.

4.2.6.4.1 Input Module

The input module should provide four facilities: 1) a form with fields for inputting the values of initializing parameters for one simulation run, 2) a dialogue for accessing a file that contains initializing values for one or more simulation runs, 3) a dialogue for specifying a file where the simulation results will be stored, and 4) commands for starting or cancelling the program execution. Having a comment field that can be printed as an explanatory note in the results output file is also useful.

Being able to initialize multiple runs from a file will save having to initialize each run and then wait for the run to end before starting the next run. Setting up multiple runs in a file also reduces the chance of input errors. Since multiple runs are usually done by systematically changing the value of an initializing variable, setting up the changing values in a file helps to ensure that each run has the correctly incremented or decremented initializing value.

4.2.6.4.2 Runtime Display

The runtime display can be as simple as a display that shows the percentage of the simulation run that has been completed, or the expected time remaining until the end of the run, or another gauge of progress. Depending on the simulation problem, having a real time display that optionally shows the step-by-step execution of the simulation can be useful. This permits checking that the simulated performance is congruent with the performance to be modelled. When experimenting with the setup of a simulation, being able to interrupt the program to view accumulated results is also useful.

4.2.6.4.3 Simulation Program Module

The simulation program module can be constructed to support either of two basic simulation approaches: 1) simulate responses to discrete events, or 2) simulate responses to the passage of fixed time intervals. The discrete event approach could be used in simulating a command and control mission scenario in which the simulation events represent events in the mission sequence. In the time interval approach, the simulation updates the system state variables at the end of each time interval. The marine traffic regulator example in Section 4.2.6.3.2.2 (page 111) was based on the time interval approach.

Figure 23 illustrates program structures for simulating the discrete event and time interval approaches. Besides the input and output modules, the figure shows the three basic modules needed to do three essential tasks: 1) generate the events or time intervals that trigger the model's responses, 2) compute responses to each event or interval, and 3) update records of the simulation state variables after the responses have been computed. Each flow chart shows a loop that runs the simulation until all the simulation events have been processed, or the preset simulation time has run out. The figure also shows that the program flows are virtually the same for the two simulation approaches.

The block for computing responses contains the system model. It could contain a set of if-then-else logic functions representing a task network similar to a SAINT type of network, or functions that generate events or changes in response to an increment of a clock time variable. The block will also usually include functions for generating values of random variables from probability distributions representing any stochastic variables that included in the model.

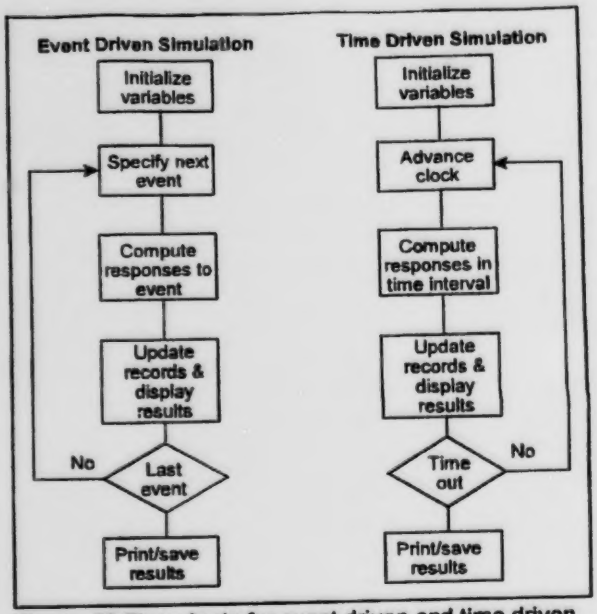

Figure 23 Flow charts for event driven and time driven simulations.

4.2.6.4.4 Output Module

An output module should provide an online display of the results from a single simulation run, or from the last run in a multiple run series. The results output file can be a text file containing, along with the results, all the initializing information needed to reconstruct each simulation run. The format of the output file need not be elaborate, but it should be constructed so that it will be readable and comprehensible long after the runs were done. It is also useful to format the output so that simulation results can be imported into other applications such as spreadsheet and graphics programs.

4.2.6.4.5 Documentation

The program should have documentation describing how to use the program, and documentation describing how the code works. If the programming language supports it, the best way to document how the program works is to create an online help file. The help file should include an overview of what the program does, describe how the program works, and describe how to perform specific program usage tasks.

Documentation of the program code can be done with comments in the program code. Even if the program is essentially a one-user application, sufficiently detailed code documentation is invaluable for debugging, and in the long term, for reusing the program code in other simulation problems. As Figure 23 should suggest, simulation programs have a fairly common structure and many common components. Therefore, if the code components are well documented, it becomes easy to build new or expended simulations once the first simulation has been built.

4.2.6.4.6 Summary of Constructing Purpose-built Simulations

The main disadvantage of a purpose-built simulation is the effort required to build it. Although writing a special purpose simulation program is no small task, modern high level languages such as BASIC make the effort much less than it once was. With a high level language, setting up the program input and output is generally straightforward. Writing the program's simulation module will usually be the most complex and time-consuming part of the job. Therefore, when considering whether a special purpose program will be worthwhile, the best way to estimate the required effort is to create a block diagram or pseudo code version of the simulation module. The effort that will be needed to create the module will be the best gauge for deciding whether or not to go ahead.

A big advantage of a purpose-built simulation is that one can closely tailor the simulation to the structure of the system to be simulated and to the data requirements of the simulation problem. If the program is properly documented, the initial simulation can be modified and developed along with the system design, and therefore, provide an ongoing analysis of the evolving system design.

4.2.6.5 The Role of Operations Statistics in Operational Sequence Analysis

This section describes three issues that affect all operational sequence analyses that rely on a model of the operation being analysed. The issues are:

1 the use of descriptive statistics from operations to estimate simulation model parameters;

2 the reliability of the simulation data, and

3 the validity of the simulation data.

The next three subsections discuss how each of these issues applies to the types of analyses that might be applied in a CCIS development project.

4.2.6.5.1 Descriptive Statistics of Operations

All simulation models require estimates of the parameters that define a particular model. To set up a SAINT task network, for example, one has to supply a value for how long it takes to do each task. To construct a service system model, one has to supply estimates of how long it takes to complete each service task, the volume of service demand, and so on. In stochastic processes, which most command and control operations are, one has to supply probability distributions representing the model's random variables. Beyond having parameter estimates, real-world performance data will be needed to verify whether or not simulation results are consistent with performance results from real systems.

In analyses of existing systems, estimates of the model parameters have to be derived from data collected in the existing system's operations. There are three sources for these data.

1 *HFE analysis results.* The task analysis specified in the CCIS HFE Requirements includes task description items that can be used as estimates of such things as the times required to do different tasks and the frequency with which different tasks are done. Mission analysis and function analysis results can provide estimates of such things as the average duration of major task sequences and the frequencies of occurrence of different mission scenarios.

2 *Operations statistics.* If the organization whose system is being analysed has collected statistics on operations, or has statistics from special studies, these statistics can be used to provide estimates of model parameters. Unfortunately, few command and control organizations collect any operations statistics, or do studies to monitor these statistics. Therefore, finding such statistics can be considered lucky.

3 *Special purpose studies.* If neither of the first two sources can provide the necessary data for parameter estimates, conducting a study to get the data needed for the estimates will be necessary. There are two possible methods for getting the data: 1) ask subject matter experts to supply the estimates, and 2) collect data samples from operations or training exercises. The first method is obviously the cheapest. The second method may be hard to cost-justify. However, it can be well worthwhile if it quantifies and clarifies the existing system's deficiencies, and the new system's performance requirements (see the discussion in Section 3.2.1.2, page 16).

In analyses of design options, particularly initial designs, one can often assume the values of some or all of the model parameters. Using assumed values of parameters is also legitimate when exploring the application of a model to a design problem. In particular, there is no point in spending time getting data to estimate a model's parameters until it is reasonably certain that the model applies to the situation at hand. Settling on the model first also ensures that the relevant data will be sought out. Although modelling a design allows more freedom in assigning parameter values, eventually, one has to show that the model provides a reasonably realistic portrayal of real world behaviour. Therefore, having realistic parameter values remains desirable even in the most preliminary of modelling attempts.

The empirical estimates of a model's parameters determine two properties that define a model's ability to represent the real world: 1) the reliability of the model's behaviour, and 2) the model's validity as a representation of the system being modelled. These properties are described in the next two subsections.

4.2.6.5.2 Reliability of Simulation Data

The reliability of a parameter estimate is the precision of the estimate, or conversely, its measurement error. Whenever a simulation is constructed, the modeller should provide an estimate of the reliability of each parameter used in the simulation, along with estimates of the reliabilities of output variables. Reliability can be expressed in many ways. In a simulation the best choice of expression is the one that best conveys the reliability of the simulation parameters to the user of the simulation results.

If a parameter has a substantial error of measurement, the effects of the error should be explored by showing how the error affects the model's results. In an analytic model, this can be done by a sensitivity analysis showing how the results vary as a function of using different parameter values. This can also be done in a simulation model if the error of measurement reflects a real-world environmental variance in the value of the parameter (rather than the modeller's lack of knowledge about the system). Such variance can be accounted for in the model by making the parameter a random variable that has the same probability distribution as the real-world effect that causes the variance.

Error of measurement in results can be estimated in either of two ways depending on the type of model being used. In an analytic model, error of measurement corresponds to the variance of the output variables. Sometimes, the mathematics of the model include functions that can be used to calculate variances. Be careful to determine if the variance values represent population or sampling variances. In an analytic model, the computations are usually based on population variances - in effect, variances from an infinite number of trials. However, the processes being modelled are finite. For example, a headquarters duty officer will handle only a finite number of messages during a duty shift or during an operation. As a result, the operation-to-operation variance in different performance variables will be greater than the population variances that may be presumed in a model of the operation. That is, the real operation's behaviour over limited times may be much more variable than the model's population variance values suggest. Consequently, one must be careful to take account of the system operating period for which the model is used to predict performance. If necessary, the variances of the model's parameters should be adjusted to reflect the sampling variance effects in the real operation. If the mathematics of the model do not provide estimates of sampling variances, constructing a simulation that will provide the sampling estimates may be necessary.

Estimating reliability in a purpose-built simulation is usually straightforward. It is just a matter of collecting data from repeated runs of the simulation, and using the data to calculate the variances of different variables.

4.2.6.5.3 Validity of Simulation Data

A simulation modeller always has an obligation to tell the user of the simulation results about the limitations of the model. The proper way to do this is by describing the model's validity. The validity of a model is the degree to which the simulation results represent the behaviour of the system being simulated. It is, in principle, easy to establish a model's validity when the model is used to simulate the behaviour of an existing system. Then, correlations between the model's performance variables and corresponding existing-system performance variables are measures of the model's validity. A valid model will produce high correlations between simulation data and corresponding existing-system data.

When the simulation is for a system that does not yet exist, there is no direct empirical way of determining the validity of the simulation model. However, validity can be shown indirectly in two ways when there is no existing system for comparison.

1 The system being simulated is similar to systems that the model has successfully simulated in the past.

2 The model's structure and components are congruent with the structure and components of the system being simulated.

When a model is an established model, a standard service system model for example, past successful use of the model is good evidence of the model's validity. This is provided that the system being simulated is clearly an example of systems to which the model has been applied in the past. Here, validity can be demonstrated by showing that the system being simulated is an example of systems to which the model generally applies.

A purpose-built simulation will often have no precedents to rely on for showing validity. In this event, validity can be shown by comparing the model's structure and its components with the structure and components of the system being simulated. Doing this is a matter of taking each component of the model and describing how well it represents the corresponding component of the system being simulated. This can include using past research results showing that individual components of the model have empirical support.

Finally, quantification of variables is an important factor to consider in the validity of any simulation of human performance. Computer simulations require that the simulation variables be mathematical quantities. Where a simulation includes variables representing human behaviour, the behaviour being represented must be not only quantifiable, but also scalable in the manner presumed in the simulation. An example of this problem appears in the simulation developed by Drews *et al.* to predict the mental workloads that would be imposed on the pilot of an attack helicopter (see description of *Micro Saint* example, page 106).

The model constructed by Drews *et al.* relied on several assumptions:

1 that Wickens' multiple resource theory of mental processes describes the workload characteristics of complex tasks;

2 that the theory can be extended to describe a helicopter pilot's attentional demands;

3 that helicopter operating tasks' attentional demands could be quantitatively scaled in the manner presumed in the simulation model, and

4 that combined attentional demands from concurrent tasks could be scaled by a simple arithmetic sum across mental processes and across tasks.

This chain of assumptions appears to have been applied in the SAINT simulation with little effort to establish its validity. While the multiple resource theory is broadly accepted for providing insight into multiple task performance, it is not the only theory. Finally, the many instances of non additive interactions between concurrent tasks [51] makes the fourth assumption, that attentional demands are additive, shaky at best.

Because simulations demand quantitative variables, it becomes a temptation to contrive quantitative performance variables just to make the simulation work. Quantifying variables is valid when supported by empirical evidence. When the quantification is not so supported, the

validity of the simulation becomes questionable. So when human performance variables are quantified in a simulation, the modeller must either show that the literature supports quantification, or conduct separate experiments to validate the quantification.

4.2.6.6 Summary of Analysis by Computer Simulation

The following points summarize the application of computer simulation analysis methods.

1 Mission, function and task analyses provide limited information about the dynamics of human performance. Simulations of doing tasks in response to time varying events can describe dynamic features of task performance; particularly features associated with the workload demands imposed on an individual or group of individuals, and the capacity of an individual or group to deal with workload demands.

2 Three simulation approaches are available for modelling human performance:

  2.1 task network models such as SAINT that are available as commercial software programs for simulating the performance of timed sequences of tasks in response to events;

  2.2 operations management service system models that provide measures of workload capacity: for example: wait times for service, times to process service demands, and utilization of resources, and

  2.3 purpose-built models that can simulate both the system and the system's environment to provide whatever measures of system behaviour the modeller needs for the simulation.

3 To be a realistic and practical representation, a simulation model has to be anchored to empirical data reflecting the modelled system's actual or likely performance. The degree to which a model represents real system performance can be represented by two properties: 1) the reliability or precision of the model's simulation results, and 1) the validity of the model's representation of the modelled system.

4.2.7 Measures of Performance and Effectiveness

Ideally, the system design requirements would specify all the performance and effectiveness measures needed to decide if the system design meets user and command and control performance requirements. The CCIS HFE Requirements does not require that the system design requirements include a specification of user performance requirements. However, the CCIS HFE Requirements does require that measures of performance and effectiveness be identified from the results of the HFE analyses, and that these measures be used in design testing.

The CCIS HFE Requirements (5.1.2) specifies that the results of the function and task analyses are to be further analysed to identify measures of user and command and control system performance and effectiveness. Eight types of measures are specified:

1 times required to execute command and control cycles for representative mission scenarios;

2 times taken to perform critical functions and tasks;

3 times to perform tasks on the command and control cycle critical path, and delays between critical path tasks;

4 times to access information required for critical tasks;

5 accuracy of performance of critical tasks;

6 accuracy of information produced by or required for critical tasks;

7 times required to perform tasks, and

8 measures of users' awareness and comprehension of tactical situation and tactical events.

The purpose of these performance measures is to provide an objective basis for conducting design reviews, and for testing and evaluating the new CCIS. The specified measures are all objective quantitative measures of either speed or accuracy of command and control performance. They are therefore also measures that are directly related to a commander's ability to exercise intelligent control over his or her forces.

The CCIS HFE Requirements (5.1.2) also specifies that a value representing minimum acceptable system performance is to be defined for each measure. In addition, if the system being developed is eventually to be evaluated by comparison with the performance of the predecessor system, data are to be collected to provide a predecessor system benchmark on each performance measure.

Ideally, the performance measures and the values representing minimum acceptable performance would be available from a prior statement of system deficiencies or system requirements. In the not unlikely event that the measures and their performance criteria are not available from a statement of requirements, they must be defined in consultation with the PM and user organization authorities. If the measures, and particularly the performance criteria, are not endorsed by the acquisition authorities, they will have little status later when the results of design tests are considered.

The performance measures can be identified by going through the HFE analysis data and marking analysis information variables that correspond to the performance measures. The performance measures should be identified keeping in mind how data will be collected for them in subsequent design testing and system trials. In particular, do not define more performance variables than it would be practical to collect data for, or variables that would be too expensive to measure.

Although CCIS HFE Requirements (5.1.2) implies that the performance measures should be identified as part of the initial HFE analysis activities, it has no explicit requirement to do so. However, though the measures will not be needed until later when design tests are done, identifying the measures will be easier when the HFE analysis data are fresh in mind.

The performance and effectiveness measures identified for the project are to be reported as part of the CCIS HFE Requirements DID titled Human Factors Engineering Analysis Report. However,

considering the important role that the performance and effectiveness measures have in design reviews and design testing, a separate document that defines the performance and effectiveness measures may be worthwhile. If the measures are expanded or changed during development, this document can be updated without having to revise the original HFE analysis report.

4.2.8 Summary of HFE Analysis Methods

The following points summarize HFE analysis methods.

1 The CCIS HFE Requirements defines three primary types of analyses: mission analysis, function analysis, and task analysis. In addition, the CCIS HFE Requirements specifies the use of system simulation methods for analysing the dynamic properties of system performance, particularly user workload demand and user workload capacity.

2 The content scope of HFE analyses covers the functions and tasks that users do in a system. Because HFE analysis results are used throughout system development and implementation, the time scope of the analyses extends from project conception through implementation. Although different analyses are more applicable at some points in a project than at others, there are no constraints on when particular analysis should be used.

3 Although there is general agreement about the meanings of the terms mission, function and task analysis, there is confusion about classifying specific analysis methods according to whether they are mission, function or task analysis methods. The following three paragraphs define the three classes of analysis for the purposes of the CCIS HFE Requirements.

   3.1 *Mission analysis* is any analysis method that describes the command and control system's mission goals, and gives an overview of the activities of command and control personnel during the three main phases of a command and control operation, that is, during the planning, directing and monitoring phases of the operation.

   3.2 *Function analysis* is any analysis method that describes the functions performed by command and control personnel as parts of functions performed by the system's hardware and software components, or as parts of function of the command and control system as a whole.

   3.3 *Task analysis* is defined specifically for the purposes of the CCIS HFE Requirements as a hierarchical critical task analysis that describes each task according to six items of information, and according to a further 10 items covering related groups of tasks. Although it requires a critical task analysis, the CCIS HFE Requirements does not preclude using other task analysis methods as supplementary analyses of users' tasks.

4 Sections 4.2.3, 4.2.4 and 4.2.5 of this CCIS HFE Design Guidelines describe analysis methods suitable for mission, function and task analysis. The mission and function analysis sections presume that the analysis data collection will be done as part of general system engineering activities. These two sections outline, first, the analysis methods applicable to command and control environments, and second, how the results of the analyses can be adapted for HFE purposes. The task analysis section presumes that the analysis will be

conducted by HFE personnel. It describes how to set up a task analysis database and how to collect the task descriptions.

5 Section 4.2.6 describes computer simulation methods for analysing the dynamic features of user performance, particularly features connected with analysing user workload capacity. The methods described include task network simulations exemplified by SAINT, operations management simulation methods, and purpose-built simulations. The section also describes ways to establish the reliability and validity of simulation results.

6 All of the HFE analyses are described in the context of the planning and monitoring phases of command and control operations. Where possible, command and control system examples are used to illustrate the use of the analyses.

7 The results of HFE analyses are the foundation for the measures of command and control performance and effectiveness measures that will be used in the HFE design reviews and design testing activities that contribute to developing a system.

These design and testing activities are described next, beginning with HFE in design.

4.3 HFE Design Methods

This section describes HFE activities that contribute directly to system design. The activities are divided so that they correspond to two system development phases. One phase is the identification of design requirements. Depending on the project objectives and the system development approach being used, this phase may also consist of producing an initial design that includes the system design requirements. The second system development phase contains the design activities that produce a detailed system design. Again, depending on the project and system development approach, this second phase could consist of anything from producing a detailed design for an interim system, to producing a system ready for production and implementation.

Each of the design phases has a corresponding set of HFE activities. HFE methods used in developing design requirements methods are described beginning with next section. HFE methods used in detailed design are described later in Section 4.3.2 (page 143).

4.3.1 HFE Design Requirements Methods

This section describes HFE methods that are generally connected with defining design requirements or developing a preliminary design. The section is divided into four parts:

1 identifying user support requirements;

2 identifying decision support requirements;

3 participating in function allocation, particularly human-machine function allocation;

4 doing tests to verify decision support and function allocation proposals, and

5 participating in iterative development activities.

These subsections correspond to the sections of the CCIS HFE Requirements that specify the HFE design requirements methods.

4.3.1.1 Identification of User Support Requirements

The CCIS HFE Requirements (5.1.4) specifies HFE responsibilities in identifying design requirements. A major implicit HFE responsibility in identifying system design requirements is to ensure that the new CCIS supports users' overall command and control tasks. Going back to an earlier discussion, a new CCIS can range from being a simple communications and database system to an advanced decision support system. If system design aims at producing the elementary CCIS, the HFE role in formulating design requirements will be limited essentially to ensuring that the system is as easy as possible to use. If the system design aims at a more advanced user support system, the HFE responsibility will extend from basic ease-of-use issues to issues of providing technological support for users' high level command and control tasks.

To reinforce its decision support aim, CCIS HFE Requirements (5.1.4) divides the process of identifying design requirements into two components: 1) identifying decision support requirements, and 2) identifying human-machine function allocations. Although the CCIS HFE Requirements treats decision support requirements and function allocation separately, in practice, the processes for identifying decision support requirements and function allocations may be the same process. This is because giving users decision support devices amounts to allocating certain portions of human decision making activity to machine elements. Consequently, the following will deal with decision support requirements by describing the nature of decision support requirements in command and control, and the types of technology that are likely to be useful in command and control decision making activities. The actual procedures for identifying and evaluating decision support technology will be described as part of the description of HFE activities in function allocation.

The following deals first with decision support requirements, and then with function allocation.

4.3.1.2 Identification of Decision Support Requirements

This section is divided into three main parts:

1 a summary of the CCIS HFE Requirements specifications for identifying decision support requirements;

2 discussion of the nature of decision support in command and control, divided into:

- 2.1 decision support for planning activities,
- 2.2 decision support for monitoring activities, and
- 2.3 the role of HFE in identifying decision Support requirements.

4.3.1.2.1 CCIS HFE Requirements Specifications for Decision Support

The CCIS HFE Requirements (5.1.4.1) specifies requirements for identifying decision support needs. The requirements are divided into two sets: 1) identification of decision support technology, and 2) evaluation of alternatives for decision support technology. The identification of decision support technology is to be done by identifying user tasks that involve cognitive activities, and then identifying technology that could improve users' performance of these tasks. Evaluation of alternative decision support technologies is to be done by comparing alternative's according to their estimated impacts on user workloads, speed and accuracy of performance, and quality of decision making.

Reporting on plans for identifying decision support technology is described in Paragraph 6.3.6.d of the DID titled Human Factors Engineering Plan. Paragraph 6.3.2 of the DID titled Human Factors Engineering System Requirements and Function Allocation Report describes how to report on the method used to identify decision support requirements, and the results of these efforts.

4.3.1.2.2 The Nature of Decision Support in Command and Control

There are no well established analysis methods specifically designed to identify decision support requirements. What is meant by the term decision support is also not well established. In software engineering, decision support usually means computer applications connected with a data base management system, and used by an organization to support higher level management decision making tasks (Sage [56][21]). However, in its most elementary form, a decision support system can be any device that helps in making a decision. Such devices can range from things as simple as a hand-held calculator, through commercial spreadsheet programs, to purpose-built expert systems. For the purposes of the following, decision support technology will be defined as any device that can support command and control decision making tasks. However, it will exclude integrated decision support systems designed to support the major components of an organization's decision making.

The reason for excluding integrated decision support systems can be explained by reference to an example. Some financial institutions have decision support systems that support decisions associated with such things as managing assets, investing deposits, exploiting economic forecasts, making corporate loans, and so on. Such a system requires a large complex database, along with an array of applications software for doing such things as computing profit and loss tradeoffs on corporate investment and loan policies. The software may include purpose-built computer models not unlike those described earlier for modelling user task performance (Section 4.2.6, page 103). Building a decision system of this sort, say as part of a moderately high level command and control system, might be a greater undertaking than building a CCIS for the same command and control system. Therefore, an integrated decision support system is not likely to form part of a

[21] The contents of this chapter are based on Sage's book: Sage, A. P. *Decision Support Systems Software Engineering.* John Wiley, New York, 1991.

CCIS, at least not at this stage of technology development. This is to say nothing of the reliability and security risks in integrating an advanced decision support system with a CCIS.

Although an integrated decision support system is not likely to be part of a prospective CCIS, there are still a variety of practical decision support facilities that can be included in the CCIS, or can exploit the CCIS platform to provide decision support. For present purposes, the decision support facilities that should be considered are ones ranging from simple automation of time consuming routine tasks to ones that provide data representations and computations for various command and control tasks. Although simple automation of routine tasks is not in itself a decision support facility, it can be an indirect support facility by releasing command and control staff time for the cognitive tasks that are the essence of command and control.

The following are examples of command and control decision support applications likely to be useful in a command and control environment. They are divided into planning applications and monitoring applications.

4.3.1.2.2.1 Planning Decision Support Applications

Command and control planning tasks are largely project planning tasks. Two general categories of applications are useful in command and control planning. One is made up of map display applications. The other is made up of the types of applications that virtually anyone planning a project could use.

Planning and Map Technology

Although each command and control organization has unique planning environments and planning tasks to deal with, virtually every military plan involves the geographic manoeuvre of forces, and the movement of supporting elements. Thus, some type of map is a fundamental medium for planning operations, and is therefore a basic decision support device. This means that for purposes of identifying decision support, the CCIS user organization's use of maps will be an obvious area to look at.

Normally command and control organizations work from three basic types of maps. One type is a main operations map showing the current deployments of forces, and other operations-related information. The other type of map is the specialist unit's map showing information specific to the nature of the specialist unit's functions: for example, a supply unit's transportation routes and supply points. The third type of map is a planning map that can be used as a sort of scratchpad map for tracing out possible operations manoeuvres or movements.

Military map media are currently in a state of transition going from paper maps to computer based electronic maps. Although the paper map is unlikely to disappear from use, electronic maps have many obvious potential advantages. With simple user commands, electronic maps can, for example, selectively display different types of information; change the format information displays;

show the movement of forces over time; provide detailed information about a manoeuvre element in response to a user enquiry, and predictively project or replay operations scenarios.

Command and control organizations fall into three technology categories according to their uses of map technology: 1) ones that rely mainly on electronic maps, 2) ones that rely on a mix of electronic and paper maps, and 3) ones that rely exclusively or mainly on paper maps. However, even if an organization already uses a good deal of electronic map technology, the technology is likely to be old, and possibly crude. Therefore, even if an organization already uses a good deal of electronic map technology, the use of maps will still be a major area of analysis for map-based decision support applications.

Other Planning Applications

The operation order is the final embodiment of a command and control organization's master plan for an operation. Producing it involves the same types work activities that any organization might do to plan a project. The four main types of devices used in these planning activities include:

1 *Spreadsheet tables* for tabulating quantities, calculating summary quantities, calculating tabulated quantities using mathematical formulae, and graphs to summarize data or show new trends.

2 *Databases* containing lists of text and data information, and reference information.

3 *Document production facilities* for producing orders, reports, briefings, memoranda and correspondence.

4 *Organization-specific and discipline-specific methods* for finding solutions to standard technical problems associated with planning.

All four of these types of devices were once manual, and are now typically mostly computer-based. Although they are not all called decision supports, they are all part of the tool set that a command and control organization can use in the decision making process that leads to a plan or operations order.

4.3.1.2.2.2 Decision Support Applications for Monitoring Activities

The decisions made during operations monitoring are generally quite different from those that go into planning, and they do not lend themselves to the types of decision support devices used in planning. Excluding routine supervisory decisions, decisions during monitoring operations consist of altering the original plan to deal with unanticipated events. These are decisions that the commander has to make quickly. While the commander may have time to confer with advisors such as immediate subordinates or staff officers, the staff will not have time for doing the type of staff work that goes into producing a plan. Consequently, the quality of the commander's decision, and the quality of the advice that advisors can provide, will depend largely on the accuracy and currency of everyone's collective mental picture of the operations situation. Except

immediately available status displays and enquiry facilities, the devices used in operations planning will be of little help in decision making during monitoring.

As this suggests, a device that supports decision making during monitoring is essentially anything that can maximize everyone's comprehension of the tactical situation. Three basic requirements must be met to maximize comprehension of the operations situation:

1 maximize the speed and accuracy with which information (messages) from subordinate elements moves up the chain of headquarters monitoring cells;

2 maximise the clarity and meaningfulness of operations status displays, and

3 ensure that personnel in headquarters monitoring cells have time to absorb and comprehend the information that comprises the operations situation.

The first requirement, yields the accurate and up-to-date information that is basic to knowing what is going on in an operation. The second requirement reflects the fact that comprehension of information is very much influenced by the manner in which the information is displayed. The third requirement reflects a common reality of information handling systems: as the workload pressure of mechanical information handling tasks rises, capacity for interpreting and understanding the information decreases. The duty officer in any headquarters monitoring cell is the commander's best early warning system for detecting that an operation plan is heading into trouble. Up-to-date accurate information presented on well designed displays is of little value if no one has time to look at it. Military history is full of examples of disasters coming from information received but not grasped.

Obviously these requirements are not decision support systems in the usual sense. Nevertheless, they are essential to effective decision making during the monitoring phase of operations. From the HFE perspective, these requirements could come under the general rubric of maximizing the ease-of-use characteristics of operations monitoring facilities. On the other hand, consideration of these ease-of-use characteristics should include consideration of impacts on decision making.

4.3.1.2.3 HFE Role in Identifying Decision Support Requirements

Although identifying decision support requirements from function and task analysis data is technically part of the HFE role, most or all of the ideas and suggestions for decision support technology will likely come from users. Users' decision support suggestions often turn up during task analysis sessions. When the user organization's subject matter experts make suggestions about decision support, or any other design suggestions for that matter, the suggestions should be recorded for later use in identifying the decision support requirements.

Whether or not users produce most of the suggestions for decision support needs, the HFE specialist should review the function and task analysis data to identify decision support possibilities that users may have overlooked. This can be done by using the function allocation approach described later in the section titled *Allocation to Maximize Technology Support for User Tasks* (Section 4.3.1.3.2.3, page 134).

Rather than identifying decision support requirements, the more important part of the HFE role will be to decide from function and task analysis data whether prospective decision support technology will have good impacts on user performance. Good impacts include reduced user workloads, higher speed and higher accuracy of performance, and better quality decision making. Section 4.3.1.3.2.3, referred to in the last paragraph describes an approach suitable for estimating the impacts that prospective decision support facilities would have on user performance.

4.3.1.3 Function Allocation

In its most general sense, function allocation is the design process of assigning system functions to the different media that are available for performing the various functions. From the HFE perspective, function allocation is generally thought of as a process of deciding which system functions will be performed by the human components of the system, and which functions will be performed by the hardware and software components of the system (Kantowitz and Sorkin [57]). In other disciplines, software engineering for example, function allocation may be the process of identifying which software modules will perform the sub-functions that make up the overall system function (Yourdon [58]).

Although function allocation in the HFE sense is widely recognized in the HFE literature, it is not so widely recognized in general system engineering. Beevis [59], for example, found in a review of a sample of five functional analyses of major military systems that none had addressed allocation of functions to the human elements of the systems. Instead, all the function allocation efforts were addressed to dividing functions among machine elements. Thus, HFE function allocation is not something automatically included in the system design process. In fact, the system design procedure may not have a function allocation process of any type. Instead, design will get underway and continue with function allocations being implicit in the design.

The most common reason for having no general function allocation procedure in a system design process is that the new system will be an instance of a well-established type of system. For example, if the new system is fundamentally another instance of the many database applications found in various organizations, the functions of different system elements will typically be taken for granted. Design will then be a matter of detailing an established basic design rather than of doing fundamentals like deciding which functions to allocate to which subsystems. Where function allocation becomes necessary is where the new system has no design precedents, and is complex enough to require serious study of the best way to define subsystems, and the best way to allocate functions to the subsystems.

Often function allocation, particularly HFE function allocation, is treated as an optional system development task. Then, the HFE practitioner has to be ready to adapt to circumstances and use whatever means are practical to ensure that the functions allocated to humans in the design make both good HFE sense. The following subsections describe, first, CCIS HFE Requirements specifications for function allocation; then, function allocation methods applicable to CCIS development. The descriptions of function allocation methods will also cover the application of

function allocation methods to identifying requirements for decision support and automation of manual tasks. The last subsection discusses the roles of HFE in CCIS function allocation efforts.

4.3.1.3.1 CCIS HFE Requirements for Function Allocation

HFE activities in function allocation are specified in CCIS HFE Requirements (5.1.4.2). This directs that HFE principles are to be used to identify information processing functions and tasks that could be allocated to automated operations, manual operations, or combinations of automated and manual operations. CCIS HFE Requirements (5.1.4.2) has two subsections. The first specifies that allocations are to be considered according to three criteria: 1) reducing user workload capacity devoted to routine or repetitive information handling tasks, 2) maximizing the timeliness, accessibility, and accuracy of command and control information, and 3) minimizing personnel selection and training costs.

The second subsection of 5.1.4.2 specifies an evaluation of function allocation options against criteria representing user performance and effectiveness. Factors to be considered include projected user performance data, estimated cost data, operating procedures constraints, and prevention or minimization of situations where human decisions are made under conditions of uncertainty, time stress, or workload stress, and personnel selection and training. This subsection also specifies the use of tradeoff studies to evaluate function allocations against different user impact factors.

Reporting on HFE function allocation activities is specified in the DID titled Human Factors Engineering System Requirements and Function Allocation Report. The DID titled Human Factors Engineering Plan refers to function allocation in Paragraph 6.36 (Human factors engineering analyses).

The CCIS HFE Requirements clearly imply that a formal function allocation procedure is to be conducted. Some of the function allocation methods that will be described in the following subsections will support a formal procedure. Others of the methods will be more appropriate for situations requiring less formal function allocation procedures, including the situation where the system development plan does not include a function allocation process.

4.3.1.3.2 Function Allocation Methods

The following describes function allocation approaches that can be used for function allocation in CCIS development:

1 allocation according to the relative capabilities of human and machines;

2 use of a formal allocation procedure that allocates functions according to a set of criteria uniquely defined for the specific allocation problem;

3 allocation based on maximising technological support for system users' tasks, and

4 supplementary techniques that, while not sufficiently comprehensive for doing a complete function analysis, may be useful for doing parts of a function analysis.

4.3.1.3.2.1 Allocation by Human-Machine Capabilities

In 1951, Fitts (described in McCormick [60], and Beevis *et al.* [18]) developed a list comparing the abilities of human and machines to do tasks ranging from detecting light or sound to exercising judgment. Successors to Fitts have generated more elaborate lists, usually to keep pace with advances in machine capabilities ([63], for example).

In theory, one can use a Fitts list to decide for a given function, which of human or machine would be better able to perform the function. However, the approach is seldom used. In practice, allocation decisions are more complicated than looking up a list to decide how to assign a function. A function allocation decision is not just a matter of deciding which of human or machine could best perform the function. An allocation decision normally has to include considerations of cost, feasibility for the design situation, organizational impacts, and the like. Therefore system developers seldom assign functions based on a one-dimensional criterion like a Fitts list (Kantowitz and Sorkin, [57]).

Another difficulty with a Fitts list approach is that the functions to be allocated are often not congruent with the human-machine dichotomy inherent in a Fitts list's metric. Many functions require a combination of human and machine cooperation. Thus, Jordan [61] has argued that instead of dividing functions between human and machine, function allocation should be a search for ways to let humans and machines cooperate.

Perhaps the main reason Fitts' lists have limited usefulness is that the items on the lists are typically self-evident generalities about the differences between humans and machine. That humans do better at inductive logic while machines do better at deductive logic is not new information to anyone. Moreover, for a given function, usually only one or two items on a list apply to the allocation decision. For example, knowing that machines can smoothly apply great force is irrelevant when deciding how to allocate a computational or decision making function. From this perspective, using a Fitts list, while having the virtue of providing a systematic approach, is usually nevertheless a case of belabouring the obvious.

The Fitts list approach also has limited usefulness for identifying decision support possibilities. Genuine decision making activities are too complex to boil down to the types of simple human-machine differences found on a Fitts list. While it may be easy to go through a task analysis of a decision making process, and identify things that a machine or computer could do better, the exercise is likely to end up being rather artificial, and shed little light on the technology that could support the decision making process.

Although the pure Fitts list approach to function allocation is seldom satisfactory, accounting for human-machine capability differences is still fundamental to the allocation decision. The formal procedures described in the next section represent approaches to function allocation that aim to

take account of differences in human and machine capabilities, and simultaneously take account of all the moderating factors that must be considered in making an allocation decision.

4.3.1.3.2.2 Allocation by Formal Rating Procedure

Meister [62] describes a formal quantitative method of function allocation that relies on weighted ratings of allocation criteria compared with weighted ratings of design alternatives. The procedure is also described in Beevis *et al.* [18]. The design alternatives can be anything from simple human-machine alternatives to alternative decision support facilities. The allocation decision criteria can be anything from simple Fitts list human-machine criteria to multiple criteria that include both relative human-machine and cost criteria. Table 5 lists the steps in the procedure. The procedure is applied only to those allocations that are not already decided by constraints such as organizational policies.

Meister's allocation procedure is essentially a Utility Theory application in which the utility of a design alternative is the sum of the products of its selection criteria weights and the corresponding values of merits on the various selection criteria. There are a variety of ways one can use to carry out this approach. Beevis *et al.* [18] (after [63]) describe an example under the heading of Function Allocation Evaluation Matrix. This method begins with the creation of a form called a function allocation screening worksheet. The worksheet is a two-dimensional table in which rows represent the functions to be allocated, and columns represent human and machine capability criteria.

Table 5 Steps in Meister's procedure for allocating functions.

1	List all possible ways to perform the function (in contrast to thinking only of how the function has been performed in the past).
2	Develop written descriptions of the alternatives.
3	List and weight the selection criteria to be used for deciding among design alternatives: cost, reliability, personnel requirements, time to implement, and so on.
4	Establish weights for the design alternatives themselves by weighting the design alternatives according to the weighted selection criteria.
5	Sum the products of the criteria weights times the alternative's weights to find the best design alternative.

Once the functions and capability criteria have been identified, each capability criterion is given a numerical weight representing an aggregate measure of the criterion's bearing on the allocation decision. The weights can include variables such as cost and availability, besides the criterion's relevance to the function being allocated. After the criterion weights have been assigned, each function is given a numerical rating on each capability criterion. The products of these ratings and their corresponding criterion weights are then summed separately for the human and machine criteria. After standardizing the summed products, the allocation then goes to whichever of human or machine received the highest aggregate rating.

In its most general sense, Meister's procedure is one of many multiple-choice decision making models that could be applied to function allocation. The basic models are described in Lehto [64]. Meister says in [5] that only rarely will a function allocation problem justify the effort required to apply a formal quantitative allocation procedure like the one just outlined. Meister says this based

only on consideration of the immediate effort applied to identifying the design alternatives, developing the criteria, and assigning their weights.

Weighting the criteria will generally involve creating subjective scales to represent the weights for any criteria that are not objective quantities such as cost. Ratings given to the functions compared with the capability criteria represent another set of at least partly subjective scales. Constructing the rating scales properly requires applying sound questionnaire item construction procedures. Section 4.4.5.3 Construction of Questionnaire Instruments, (page 210), describes the procedures for constructing such scales.

Care must also be taken to establish the reliability and validity of each scale as discussed earlier in Section 4.2.6.5.2 (page 118), and Section 4.2.6.5.3 (page 119) in connection with the reliability and validity of variables used in a simulation model. Thus, establishing the reliability and validity of the types of weights and ratings used in a function allocation exercise may itself be no small matter. If the validity and reliability of the variables used in the exercise are left undetermined, the whole procedure rests on a complicated and unsubstantiated chain of psychophysical scaling assumptions. Section 4.4.5.4 Construction of Expert Panel Decision Instruments (page 223) describes the factors that must be dealt with to obtain useful results from a multi-attribute decision making process like Meister's function allocation procedure.

4.3.1.3.2.3 Allocation to Maximize Technology Support for User Tasks

The function allocation approach described in this section is not among the conventional approaches described in the literature. It is, however, consistent with much of real-world system development practice, and it is consistent with the CCIS HFE Requirements specifications for function allocation. It is also particularly consistent with how decision support technology is identified in practical applications [56].

Traditional hardware function allocation came from a point of view where humans could be treated as subsystem alternatives to machine subsystems. A different point of view might be better for function allocation in a CCIS development. In a command and control system, the CCIS is a support subsystem for the humans who comprise the command and control system. Thus, in CCIS function allocation, a function allocation is not a question of whether human or machine should perform a function. It is more a question of whether or not a machine can take on or support a human function.

In a sense, looking at function allocation as finding ways to support command and control tasks avoids the function allocation question altogether. This is especially so if the system requirements are defined as needs for improved performance of command and control tasks. Then, instead of allocating functions, the designer looks for machine systems that could improve the performance of command and control tasks. This approach is consistent with the role of HFE in function allocation specified in CCIS HFE Requirements (5.1.4.2) which says that HFE principles are to be used to 'identify information processing functions and tasks that could be allocated to automated operations, manual operations, or combinations of automated and manual operations.'

Although a great deal of information system development is based on finding ways to improve an organization's performance by giving personnel automated support, there is no standard function allocation procedure for taking this approach. The following suggests a procedure that could be used. The procedure is essentially what is usually done in practice, with the addition of some steps aimed at validating the HFE wisdom of design choices made along the way. In the abstract, the procedure will not be logically different from the class of formal function allocation procedures described in the last section (Section 4.3.1.3.2.2).

The suggested procedure is based on taking the function and task analysis results for the predecessor command and control system, and going through the functions and tasks to identify functions that could be supported by new or improved technology. The steps in the procedure are the following.

1 From the predecessor system function and task analysis results, screen out functions that would be candidates for computer support.

2 For each candidate function, identify candidate technologies for providing the computer or other technological support.

3 Identify the dimensions on which the merits of candidate technologies will be compared.

4 Construct a process for aggregating the merits of each technology into an overall order of merit.

5 Use the overall orders of merit to select the technology to employ for each function.

The steps in this procedure are parallel to those in the Meister's function allocation procedure (see Table 5, page 133). However, the procedure outlined here does not require the use of any quantitative metrics for estimating the merits of candidate technologies, or for comparing candidate technologies. Neither does it forbid the use of quantitative metrics when there are data to justify quantification.

The possible dimensions for evaluating candidate technologies are bound to include a mix of variables representing system development considerations and impacts on user performance. System development considerations would likely include such things as development cost, reliability and maintainability, technical feasibility, and so on. The user performance dimensions can be defined simply by compliance with the function allocation or decision support requirements sections of the CCIS HFE Requirements. The CCIS HFE Requirements (5.1.4.2.1 and 5.1.4.2.2) specify three dimensions for what amount to criteria for identifying candidate functions:

1 potential for minimizing the workload capacity that users must devote to routine or repetitive tasks;

2 potential for maximizing the currency, accessibility, and accuracy of command and control information, and

3 potential for minimizing personnel selection and training costs.

Dimensions for evaluating candidate technologies can be defined by compliance with CCIS HFE Requirements (5.1.4.2.2). This defines the relative merits of proposed function allocations by reference to measures of performance listed in CCIS HFE Requirements (5.1.2), that is,

1 times required to execute command and control cycles for representative mission scenarios,

2 times taken to perform critical functions and tasks,

3 times to perform tasks on the command and control cycle critical path and delays between critical path tasks,

4 times to access information required for critical tasks,

5 accuracy of performance of critical tasks,

6 accuracy of information produced by or required for critical tasks,

7 person hours or minutes required to perform tasks, and

8 users' awareness and comprehension of the tactical situation and tactical events.

There are three choices for scaling these variables for the purposes of a particular CCIS:

1 define the variables as dichotomous checklist items in which a candidate technology would get a yes or no rating on each dimension;

2 define the variables as rank order variables in which each candidate is ranked relative to other candidates, or

3 define a numeric rating scale for each variable so that a candidate's merit on a dimension would be a rating value.

At an early stage in design, the first approach is probably the most valid and convenient. From a measurement point of view, the three scaling definitions represent increasing degrees of measurement precision. Each is useful to the degree that its measurement precision is consistent with the precision to which the technology candidates can be evaluated. If the precision of the evaluation (by whatever means the evaluation is made) is necessarily low, using a finely scaled metric is pointless. In the early stages of design, making precise evaluations of the merits of candidate technologies will not likely be possible. Therefore, using a checklist or rank ordering metric will likely be the most appropriate way to evaluate candidate technologies early in design. Later a more precise evaluation metric might be warranted if data become available to support the precision, or if system criticality or costs make it important to evaluate technology options as precisely as possible.

The procedure just outlined can be done as general system engineering exercise to identify decision support possibilities or allocate functions. Alternatively, the procedure can be conducted as a separate HFE exercise to estimate the HFE impacts of different technology options. The latter approach permits compliance with the CCIS HFE Requirements without imposing a formal function allocation procedure on the rest of the system engineering effort. The aim of the CCIS HFE Requirements specifications for function allocation is to ensure that the HFE implications of allocations are given due consideration. Whether this is done through a general function allocation process or done as a separate HFE process does not matter.

4.3.1.3.2.4 Other Function Allocation Procedures

Beevis *et al.* describe some additional function allocation procedures that may be useful in some circumstances. These are briefly described in the following paragraphs.

Requirements Allocation Worksheet.[22] This is a predefined form containing a matrix in which rows represent functions. The columns represent facilities and equipment specifications, personnel requirements for operating tasks, task performance requirements, training requirements, and so on. The requirements allocation worksheet is not so much a function allocation procedure as it is a way of systematically relating function requirements to system design requirements.

Critical Design Requirements [65]. This is a procedure that identifies the critical operating variables for each operating task, and in turn, the critical design requirements that affect these variables. The procedure is intended for equipment work space and interface design. Since the procedure is not narrowly defined, it can be incorporated into other allocation procedures, or applied to isolated function or task sequences.

Review of Potential Operator Capabilities.[23] This is a NATO requirement specifying a review of the performance capabilities that will be needed by the system's human components to support different system functions. It does not specify the capabilities to be considered, or a documentation format of the review. The procedure is apparently not widely used.

Design Option Decision Tree. Applying this technique for HFE purposes consists of constructing decision trees representing design options, and then for each decision point, listing the HFE advantages and disadvantages of the design options. Beevis *et al.* give an example of applying the procedure to ship bridge design (from Beevis [66]).

More variations on function allocation techniques can be found in the HFE literature, particularly in the older edition of the *Handbook of Human Factors* (Kantowitz and Sorkin [57]). The more recent *Handbook of Human Factors and Ergonomics*' function allocation chapter is devoted almost entirely to dynamic function allocation. This is the name for a system's capability to switch functions dynamically either to human or machine depending on operating conditions. Discussion of this will be left to a later section.

---

[22] Beevis *et al.* describe this procedure by reference to:
USAF (1971). Data Item Description – Requirements Allocation Sheets. DI-S-3605/S-127-1. Washington D.C.: US Air Force Systems Command.
US Defense Systems Management College (1990). System engineering management guide. Washington D.C.: US Government Printing Office.

[23] Beevis *et al.* describe this procedure by reference to:
NATO MAS (1991). Application of human engineering to advanced aircraft systems. STANAG 3994 AI (Edition 1). Brussels: NATO Military Agency for Standardization.

4.3.1.3.3 Roles of HFE in Function Allocation

The section corresponding to this one in the CCIS HFE Requirements User Manual (4.2.2.4) outlines four tasks that HFE specialists can do in function allocation.

1 Furnish estimates of human capabilities for performing each function to be allocated.

2 Identify human performance criteria to be used in selecting design options representing different function allocations: for example, human capacity to perform the function, performance reliability, impact on user workloads, implications for personnel complement costs, and costs associated with knowledge, skills and training requirements.

3 Evaluate design options against the human performance criteria defined for the function allocation procedure.

4 Design, or participate in the design of the function allocation procedure to be used in the system design process.

The first three tasks can be fulfilled by meeting the CCIS HFE Requirements specifications for function allocation by contributing to whatever function allocation procedure is used in the System Engineering Plan: for example, by participating in one of the allocation procedures described in Section 4.3.1.3.2. The fourth task can be fulfilled by contributing expertise in behavioural measurement to constructing and conducting a formal function allocation procedure like that described in Section 4.3.1.3.2.2.

Constructing a reliable and valid formal function allocation procedure has several technical requirements that an HFE specialist can satisfy.

1 Define criteria weights and measures of merit variables to minimize measurement error owing to such things as inappropriate choice of units of measurement, error variance, and so on.

2 Ensure that each subject matter expert's ratings on different variables are weighted to reflect his or her qualifications to make judgments about each variable.

3 Construct a rating procedure and data collection instruments that ensure that all subject matter experts can provide reliable and unbiased ratings.

4 Analyse the data from the allocation procedure to provide intelligible findings balanced by a fair assessment of the reliability and validity of the results.

A formal function allocation procedure is essentially an evaluation of alternative imaginary systems. As such, constructing a valid and reliable function allocation procedure requires all the same skills required in constructing the system design tests that will be described later in Section 4.4 (page 184). As a final word on formal function allocation, possibly the greatest service the HFE specialist can provide is to discourage the use of too elaborate numerical procedures that are superficially precise and objective, but notoriously unreliable and difficult to interpret.

4.3.1.3.4 Summary of Function Allocation

The following points summarize the HFE use of function allocation in CCIS development.

1 For HFE purposes, function allocation is the process of deciding for each system function, which of human or machine would be the best design option for performing the function. Notwithstanding this definition, actual system analysis and design often treat function allocation as deciding among machine options for performing functions - essentially leaving humans to do whatever cannot be done by machine.

2 The CCIS HFE Requirements specifies a function allocation process in which allocations are evaluated according to the basic criteria: 1) reducing user workload capacity devoted to routine or repetitive information handling tasks, 2) maximizing the timeliness, accessibility, and accuracy of command and control information, and 3) minimizing personnel selection and training costs. The CCIS HFE Requirements specifies additional variables to be considered in allocation decisions: 1) estimated cost data, 2) operating procedures constraints, 3) prevention or minimization of situations where human decisions are made under conditions of uncertainty, time stress, or workload stress, and 4) personnel selection and training impacts.

3 CCIS HFE Requirements provisions for identifying decision support needs can be met by treating the search for decision support facilities as a function allocation activity.

4 Three basic function analysis approaches are available for making HFE contributions to function allocation decisions.

4.1 *Fitts List.* This consists of evaluating human-machine design alternatives according to comparisons of human and machine capabilities. Though fundamental and widely known, the approach is seldom used because it oversimplifies the allocation decision, and ignores other variables that are relevant to the allocation decision.

4.2 *Formal Function Allocation Procedure.* This approach consists of first identifying alternative design options for the various system functions, and then identifying a set of allocation decision criteria. These criteria include any variables germane to the decision: costs, system performance outcomes, Fitts list capabilities, and so on. Each criterion is then assigned a weight representing the criterion's judged importance for the allocation decision. The allocation choices are then rated according to the decision criteria - usually by subject matter experts. The ratings are then weighted according to the criteria weights, and aggregated into an overall measure of each allocation candidate's merit. The procedure is systematic, flexible, and it can be constructed to any level of quantification required. However, a complicated set of criteria, weights, and rating procedures can make the procedure vulnerable to reliability and validity problems.

4.3 *Allocation to Maximize Machine Support for User Tasks.* This procedure consists of going through users' functions and tasks and identifying ones whose performance could be improved by machine support. Options for support technology can be evaluated by adapting a formal function allocation procedure, or by doing an independent HFE evaluation of each option according to the user performance criteria specified in the

CCIS HFE Requirements. The procedure is well suited to identifying and evaluating decision support facilities. The procedure is also well suited to general command and control applications, and allows compliance with the CCIS HFE Requirements function allocation specifications without imposing a function allocation process on the general system engineering program.

5 Along with contributing HFE knowledge to individual function allocation decisions, HFE specialists can contribute knowledge of measurement processes to the design of the function allocation procedure, and to the analysis of the function allocation results.

Depending on the system development approach being taken, identification of decision support facilities and function allocation activities produce either a set of design requirements or an initial system design.

4.3.1.4 Tests to Verify Decision Support and Function Allocation Proposals

The CCIS HFE Requirements (5.1.5) gives provisions for conducting experiments or tests to verify user support proposals. Experiments or tests are to be conducted when the impacts of proposed user support facilities cannot be estimated from HFE literature sources. The main empirical approaches for verifying proposals are

1 using prototypes to measure the impacts of proposed user support requirements and function allocations;

2 conducting experiments with subjects representative of users to measure the effects of proposed user support facilities;

3 using computer simulation methods to estimate the impact of proposed facilities, and

4 conducting demonstrations with storyboards, mockups or other representations to give users a basis for estimating the impacts of design proposals.

Advice on conducting experiments or tests based on prototypes, tests performed on users, and using storyboards will be described in Section 4.4 (page 184). The use of computer simulation methods has already been described in Section 4.2.6 (page 103). Since the techniques for conducting experiments are generally independent of the purpose of experiments, the contents of the referenced sections will serve for setting up tests of user support proposals. They will not necessarily serve for deciding when a user support proposal should be empirically tested.

The CCIS HFE Requirements presume that, as a rule, HFE literature and HFE professional judgment will provide sufficient grounds for estimating the impacts of prospective user support facilities. Although making precise or quantitative estimates of impacts is seldom possible, precise quantitative estimates are seldom needed. All that is usually needed is to know that a proposal will yield materially better performance than the predecessor system or competing proposal would.

A decision to do an empirical test is essentially a matter of the collective confidence that everyone, including the HFE specialist, has in an estimate of impact that is based on existing HFE

knowledge. How much confidence is needed, or is possible, depends almost entirely on the circumstances of the project. Although there are no absolute criteria for deciding when a user support facility should be tested, the following factors are usually worth considering.

1 Prior HFE knowledge is insufficient for estimating the impacts of the proposed design.

2 The proposed design supports an especially critical command and control task, or imposes other significant risk on the success of the development project.

3 An expensive detailed design effort would be required to get to a point where the design could be empirically tested.

If a design proposal is new technology, the lack of HFE precedents for the technology may leave little choice but to do some empirical testing. Alternatively, design must continue in the dark about impacts on user performance. However, if the proposal is new technology, chances are that its other technical properties will need testing also. If so, HFE testing can be included in this testing.

The second decision criterion is almost self-evident: that is, the more critical the task that the design proposal supports, the more important is it to test the proposal.

The expense of the design effort that will go on before a proposal can be tested is probably the most important of the three decision criteria. Nearly all system development plans require design testing as part of the acquisition authority's acceptance procedure. Therefore, eventually a design proposal will be empirically tested. Thus, estimates of a design proposal's impacts are essentially estimates of the results of the system tests that will be done later. The risk here is not so much that the design proposal will fail when the system is implemented, but that a large design effort may be wasted on something that proves not to work. If this risk is substantial, then it will be worth investing in testing the proposal in advance. Conversely, if a proposal's design effort is modest, or the proposal can be tested early in detailed design, prior testing is unnecessary.

Finally, the three criteria for deciding the need for prior testing are essentially additive. That is, prior testing is more justified if all three criteria are true than it is if only one or two are true.

4.3.1.5 Iterative Development of Design Requirements

Approaching the detailed design phase through an iterative process in which a series of small sub-projects is used to produce increasingly refined design is a common in system engineering practice. The CCIS HFE Requirements (5.1.6) allows for iterative development approaches. However, since HFE practice has nothing that is equivalent to iterative development in the general system engineering sense, the 5.1.6 provisions are limited to requiring that the HFE Plan include estimates of nature and extent of the iterations that will comprise development of the system design requirements.

Although iterative development approaches have the merit of allowing a degree of trial and error in developing requirements in unexplored development territory, their inherent uncertainty makes

them difficult to manage and control. Likewise, their associated HFE activities may be difficult to plan and manage. The HFE implications of iterative development are also discussed in the CCIS HFE Requirements User Manual (5.3.2.2.1.4 and 5.4.2.3). Section 5.3.2.2.1.4 discusses the technical considerations for applying HFE to iterative development approaches. Section 5.4.2.3 discusses how iterative development affects HFE project management.

Beevis[24] points out that although the HFE activities associated with iterative development can be hard to plan or manage, two rules-of-thumb can be used. First, if the system engineering effort has anticipated a specific number of iterations then the HFE activities can be based on that. Second, the HFE plan can anticipate the minimum level of effort required for each major operator-machine interface. Nielsen [67] recommends that HFE specialists plan for least three iterations of each operator-machine interface. DCIEM experience suggests that story boarding followed by three iterations using prototypes of some sort is an effective approach.

Properly done, an iterative development process is a planned series of sub-projects, each intended to achieve a specified development objective, or answer a specified research and development question. The plan for each sub-project includes contingencies specifying the possible outcomes of the sub-project, and what will be done in the next sub-project depending on the outcome of the preceding project. At the end of each sub-project, the series plan can be modified to take account of what has been achieved so far. In this type of iterative process, the HFE issues in each sub-project can be treated much the same as they are treated in any other project.

On occasion, the term iterative development may be used to describe what amounts to a trial and error hunt for an elusive design solution. The process may consist of trying out successive design solutions until one that works is found. The process seldom includes a plan beyond the completion of each attempted design solution. Clearly, this is not effective design behaviour. Two things can cause it to happen.

One cause of trial and error design is pre design analysis that was inadequate and did not lead to a set of detailed design requirements. Having completed the analyses without coming to an explicit set of detailed design requirements, the developer presses on into design and hopes for the best. What the developer should do when this happens is go back and complete a proper requirements analysis.

The second cause of trial and error design, one most likely to occur in software development, comes from the ready availability of prototyping tools and high level programming languages. With convenient prototyping available it is tempting to do a hasty initial requirements analysis, and then start prototyping designs. The prototypes are shown to users to see what the users like. With progressive user-guided refinements, the structure and features of the prototype will eventually represent a detailed design for the target system. This procedure may work for small systems, but it is risky when applied to larger systems. While the process may start with a simple prototype with limited functionality, getting the prototype to a stage where it has realistic

---

[24] Personal communication citing DCIEM experience.

functionality may become a major programming effort, with increasing delays between prototyping efforts and user tests.

Trial and error development comes from inadequate analyses of system requirements. HFE specialists cannot single handedly prevent this situation. However, they can avoid contributing to it happening by ensuring that the HFE analyses are done thoroughly, and that their results are sufficient to support the design effort. If the criteria for deciding when a task analysis is adequate (see Section 4.2.5.4.1.6, page 92) have been met, the HFE analyses are very likely adequate.

4.3.2 HFE in Detailed Design Activities

The CCIS HFE Requirements specifies essentially two requirements in detailed design. One is a requirement to maintain a function and task analysis that represents users' functions and tasks in the system design. The other requirement is a two-part system design requirement, the parts of which are 1) to incorporate HFE principles into the system design, and 2) to review system design representations against HFE criteria, and recommend corrective action for design properties that do not meet the HFE criteria. Meeting the requirement for maintenance of a current function and task analysis is straightforward, and will be dealt with shortly in Section 4.3.2.2 (page 145). Meeting the requirement for incorporating HFE principles into system design is not so straightforward.

The CCIS HFE Requirements explicitly delegates the design review function to "personnel assigned human factors engineering responsibility." It does not delegate responsibility for incorporating HFE principles into design to anyone in particular. Obviously the design team personnel have ultimate responsibility for incorporating HFE principles into design. However, this does not necessarily relieve HFE personnel from having any responsibility for incorporating HFE principles into the system design.

How HFE specialists go about influencing design in a particular project will depend on project organizational variables, and on the mix of knowledge and experience that comprises the project team. In short, each project team has to evolve its own working procedure for incorporating HFE principles into design. Although no one procedure will likely work for all project teams, considering it the problem by reference to what might stand as an extreme scenario will be helpful. This is to make designers solely responsible for design, and make HFE personnel solely responsible for HFE design review.

Under this scenario, designers develop designs. HFE specialists review the designs and identify any design corrections needed to make the designs comply with HFE principles. Unless the designers have good HFE qualifications, this approach leaves designers guessing about what makes up an HFE design principle, and then left waiting to see what comes out of the HFE design review. This is a hit-and-miss approach that, despite its organizational tidiness, is bound to waste both designers' time and HFE personnel time. Also, designers will not respond positively to continual after-the-fact HFE critiques of their work. A better approach is to find ways of giving designers as much HFE help as possible when they execute design ideas, and by that minimize the

proportion of HFE design influence that has to come from the results of design reviews. This is the approach that the contents of the remainder of this section will aim at supporting.

To support a cooperative approach to incorporating HFE principles into design, the remainder of this section treats HFE design activities as a set of HFE tasks. Some of these tasks are direct reflections of the specifications in the CCIS HFE Requirements (5.2). Other tasks are not so much required tasks, but more so ways in which HFE specialists can divide up their work to support and influence the design effort. The tasks are the following.

1 Maintain the currency of the function and task analysis data.

2 Take account of organizational factors that control HFE influence on design.

3 Identify and help in applying relevant HFE standards and guidelines.

4 Assist in translating HFE data into design.

5 Assist in human-computer interface design.

6 Assist in equipment, work station and work space design.

7 Assist in decision support applications design.

8 Assist in automated functions design.

9 Assist in the development of user manuals.

10 Conduct HFE design reviews.

These tasks will be described in subsections following next subsection which describes the CCIS HFE Requirements specifications for HFE activities in detailed design.

4.3.2.1 CCIS HFE Requirements Specifications for Integrating HFE in Design

The requirements for applying HFE principles to system design are given in CCIS HFE Requirements (5.2). The introduction of 5.2 outlines the overall HFE responsibility in system design.

CCIS HFE Requirements (5.2.1) describes the requirements for keeping the function and task analysis up to date with the system design. CCIS HFE Requirements (5.2.2 to 5.2.6) describe, respectively, design requirements for human-computer interfaces; equipment, work spaces and work stations; decision support applications; automated functions; and user manuals. Each of these paragraphs specifies requirements both for integrating HFE principles into the design, and for reviewing designs against HFE design criteria.

The DID titled Human Factors Engineering Plan contains the requirements for reporting plans for HFE design activities. Requirements for reporting the results of the function and task analysis work are given in the DID, Human Factors Engineering Analysis Report.

The DID, Human Factors Engineering Design Approach Document, gives the requirements for reporting the HFE design approach used in meeting the requirements described in CCIS HFE Requirements (5.2.2 to 5.2.6). The HFE design approach is made up of the HFE principles that were applied to the designs of human-computer interfaces; equipment, work spaces and work stations; decision support applications; automated functions; and user manuals. The DID titled Human Factors Engineering Design Review Report gives the requirements for reporting the results of design reviews corresponding to the HFE principles that were applied to the different components of the system design.

The following sections describe the HFE tasks that need to be done to meet these requirements.

4.3.2.2 Maintain Currency of HFE Analyses

Having a task analysis describing the tasks that users will do in the system design is indispensable to all the HFE activities that contribute to the HFE quality of the system design. Procedures for keeping the task analysis up to date during system design have already been described in Section 4.2.5.4.2 (page 95), as part of the general section on task analysis (Section 4.2.5, page 73). Making use of the function and task analysis data in other HFE design tasks will be described as each design task comes up.

4.3.2.3 Organizational Factors in Applying HFE Data to Design

The CCIS HFE Requirements demand the application of HFE principles to the system design. The introduction to this section suggested that doing this by leaving designers to their own devices, and just accepting or rejecting design proposals according to HFE design criteria was an inefficient way to achieve good HFE design. A more efficient approach is to provide designers with as much HFE design advice as possible during design work. This will make the HFE design review a final check on design acceptability rather than a primary medium for achieving sound HFE design. More efficient as this approach may be, there is a certain amount of tradition in HFE suggesting that this approach is not easy to put into practice.

The difficulty in integrating HFE principles into design can come from a combination of the designers' resistance to HFE and the HFE specialists' lack of insight into design processes. Whether or not these two sets of attitudes are altogether valid in a specific project, the HFE specialist will be wise to assume that they are valid to at least some degree, and work to overcome them. Overcoming these attitudes amounts to an unwritten HFE design task. The remainder of this section describes how to do this task.

Things that can be done to help integrate HFE principles into design can be described by reference to Meister's writing about the use of HFE in design. Meister [5] (Chapter 3) comes to five conclusions about system design practice that represent barriers to incorporating HFE data into design.

1 Designers are indifferent to human factors, do not know much about human factors, and give human factors low priority in design;

2 These attitudes are reinforced by beliefs and assumptions that are mostly wrong or simplistic.

3 Designers are conservative, not completely systematic, and work on the principle of least effort.

4 HFE practitioners often react to design solutions rather than propose design solutions because they believe that their role is to evaluate design options.

5 Because of designers' attitudes, HFE practitioners spend much time trying to indoctrinate designers into belief in the value of HFE.

Though jaundiced, and possibly less true now than in 1989 when Meister published them, these conclusions still have a certain amount of validity - enough validity at least, that they provide a tactical framework for things that the HFE specialist can do to contribute to the design process.

Designers' indifference to HFE usually comes from not knowing much about HFE, and from having never had to pay attention to HFE in a design project. The CCIS HFE Requirements package is specifically intended to preclude indifference to HFE in design by making HFE requirements part of the design requirements. The CCIS HFE Requirements gives acquisition authorities a detailed procedure for inserting an explicit HFE component into a CCIS development project. The CCIS HFE Requirements User Manual provides the acquisition authorities and contractor managers with essentially a cookbook for managing the HFE component of the system development plan. This CCIS HFE Design Guidelines gives HFE specialists additional support for integrating HFE into design.

One thing the CCIS HFE Requirements cannot do is specify tailoring of the HFE process for an individual project. In particular, the CCIS HFE Requirements cannot prescribe how to incorporate HFE into the day-to-day design process. The HFE specialist has to take a large degree of responsibility for this. The following suggests some ways to exercise this responsibility.

One thing the HFE specialist can do is to recognize and deal with Meister's third conclusion: that designers are conservative, not completely systematic, and work on the principle of least effort. Designers are not necessarily conservative and parsimonious with effort by nature. Nevertheless, the realities of project economics generally demand conservative approaches that reduce the chances that things will not work, and reduce the chances of running out of budget before development is completed. HFE specialists are bound to be subject to the same pressures. The constraints that lead to conservative design and thrift in effort suggest that HFE specialists need to concentrate on providing design ideas that are demonstrably workable, and requires minimum effort to incorporate into the system design.

Chapanis [6] particularly emphasises the need for giving designers practical advice. Chapanis says that the HFE specialist must give designers explicit prescriptive advice on what needs to be included in a design execution to make the design effective for users. Similarly, HFE specialists must provide precise quantitative predictions about how well users will perform on any proposed design option. Designers cannot be expected to have much time for the HFE specialist who

serves as a talking HFE textbook giving advice as rules and principles. Most designers do not much care what is in the HFE textbook. They care about efficient, credible and workable solutions to design problems.

Meister points out that design is often not systematic. This is inevitable as much as design is a creative process. The reality of system design is that reliable procedures for translating system requirements into a system design are hard to find. System engineering has many methods for analysing systems and identifying requirements, but few credible procedures for synthesising requirements into a whole-system design that will satisfy the requirements. There is no system design demon that lets one dump requirements into a hopper at the front end, and take delivery of the completed design at the back end. At some point, the designer has to make a creative leap and settle on a design approach or design concept. This will not be a systematic process.

Parenthetically, the lack of methods for translating analysis into design is perhaps why development projects often seem to stall after the requirements analyses have been completed. When projects stall at this point, the response is often to do more requirements analyses in hopes that a design notion will materialise. The additional analyses seldom help. When a project stalls, or begins to regress into more analysis activity, the HFE specialist can do little but resist becoming part of the problem. The HFE specialist always needs to avoid the temptation to do more detailed HFE analyses than necessary. In any analysis, analysis effort produces diminishing returns once the initial information has been collected. For example in task analysis, once reasonably comprehensive narrative task descriptions have been collected, additional analysis often produces little useful information. Often these narrative descriptions will be sufficient to suggest most of the design requirements, or actual system design features. Increasingly detailed analyses may only be worthwhile after design gets underway: for example, to determine design specifications for critical tasks. Although the CCIS HFE Requirements potentially specifies much detail in analyses, it does not encourage analysis effort that will not clearly contribute to good design.

Meister's conclusion that HFE practitioners tend to react to design solutions rather than propose design solutions because they believe that their role is to evaluate design options is probably common experience. Sometimes this attitude reflects a scientific academic background that emphasizes hypothesis testing based on comparing competing hypotheses. In other cases, as Meister says, reluctance to propose design solutions reflects reluctance to invade the designer's territory, or reluctance to overstep the bounds of HFE responsibilities. No doubt, some designers are territorial and discourage outright design suggestions. Faced with such individuals, the HFE specialist has little choice but to be reactive. However, the existence of individual designers with such attitudes is no excuse for assuming that all designers have this attitude. The fact remains that often the best way to convey design advice is by a design proposal that can either be adopted, or be used to explain the rational for the advice.

Reluctance to offer prescriptive design advice is similar to the reluctance that Chapanis [6] says HFE practitioners have about predicting what will happen if a particular design option is adopted. Instead, of making predictions, many HFE practitioners prefer to withhold or hedge an opinion until the design proposal can be tested as a prototype or other design representation. Undeniably,

HFE knowledge does not always allow a firm prediction of how a design will work. Then, a forthright opinion that testing is needed is proper. On the other hand, continual refusals to predict how things will work will soon discourage designers from asking for advice. As Chapanis points out, engineering is essentially a business of predicting how something will work. It is not, like much of behavioural science, a business of finding out how things work after the fact. Thus, the HFE practitioner who sticks to the view that prediction is not a legitimate HFE role will soon find his or her advice, if it is asked for, relegated to the back pages of the system engineering reports.

In his final conclusion, Meister says that, because of designers' attitudes, HFE practitioners spend much time trying to indoctrinate designers into belief in the value of HFE. This is a generality that is difficult to confirm, or even interpret. Whatever the truth of the conclusion, a corollary is that spending time on indoctrinating designers into belief in human factors is probably wasted time. A designer's good opinion, not to mention knowledge, of HFE will come from positive experiences of HFE's usefulness in solving design problems.

Although much of the writing on getting HFE into design is about attitudes, it does suggest concrete things that HFE specialists can do to make HFE accessible to the design effort.

1 Try to ensure that the project team organization provides direct access between designers and HFE specialists. Designers should have access to HFE specialists to discuss design issues as the issues arise. Do not make periodic review meetings the sole means of liaison between design and HFE personnel. Review meetings are effective for monitoring progress and resolving big issues, but they are not effective for providing ongoing HFE support for design activities.

2 HFE specialists must be prepared to provide prescriptive design advice, provide predictions about how users will perform on proposed designs, and be generally forward-acting and constructive in exercising influence on design. Tactically, the HFE specialist needs to avoid getting into the position of being an after-the-fact design critic. Unless they are categorically unwelcome, design suggestions or examples are efficient and effective ways to convey HFE advice. Direct participation in design activity can be useful when the HFE specialist has the knowledge and skills needed to make significant contributions.

3 The HFE specialist must control the natural tendency to rely too much on analysis as a basis for contributing to design. Because analysis effort obeys the law of diminishing returns, every extension of detailed analysis effort should be evaluated for its likely contribution to the purposes of the HFE Plan. While analysis data are essential for understanding system users' needs, too many data can have just the opposite effect.

The preceding suggests organizational strategies for creating a constructive environment for meeting the system design objectives of the CCIS HFE Requirements. The next section discusses the mechanics of exercising HFE influence on design.

4.3.2.4 Measures for Making HFE Data Available to Designers

This section describes mechanisms for making HFE data conveniently available for use in design activities. One way or another, designing a system to improve a user's work consists of using knowledge of the user's tasks to devise tools that fit the user and make it easier for the user to do the work. Thus the day-to-day work of a designer is combining the information about a user's tasks with information about what makes a tool fit for use, and then drawing up plans for the tool that will best meet the user's needs. To design a fit and useful tool, the designer needs ready access to both the HFE analysis data and relevant HFE knowledge about user capabilities.

If the designer cannot have an HFE specialist constantly at his or her side, ready access to HFE information has to be provided by other means. The following are measures that will help to give designers convenient access to HFE knowledge and data.

1 Provide mechanisms for direct access to HFE analysis data and relevant HFE knowledge.

2 Be available for consultation to explain, expand or clarify HFE analysis data

3 Participate in design concept development sessions.

4 Provide operations scenarios for testing design ideas.

5 Provide predictions of user performance.

The following subsections explain these measures and suggest ways to make them effective.

4.3.2.4.1 Convenient Access to HFE Information

Designers need two sets of media for access to HFE information: one set for HFE analysis data, and one set for HFE reference information. The next two subsections describe each of these two needs separately.

4.3.2.4.1.1 Access to HFE Analysis Data

Having HFE analysis results does not guarantee that they will be used. An illustration of this is a system development project that made little use of analysis results although the project started with a thorough analysis of users' functions and tasks[25]. Even after the system design was evidently not satisfying design requirements, the analysis results remained largely ignored. One likely reason that the analysis results were largely ignored was that they were contained in a large compendium that would be difficult for a designer to unravel. Additionally, the people who did the analysis were no longer available to the designers.

Comprehensive HFE analysis data, like any other system analysis data, can fill many pages with detail that is difficult for a designer to distill into information that is useful at different stages of

[25] Author personal communication.

design. Early in design, say at the concept stage, concise narrative task descriptions, broken down into no more than one or so levels of sub-tasks, are usually adequate for the designer's purposes. At later stages of design, more detailed information on specific sets of tasks becomes useful: for example, information on specific task variables such as times to complete tasks. As useful as detailed task information may be, it will not be used if the designer judges that the effort of extracting it from HFE reports will be greater than the payoff in expected design value. As this implies, the HFE analyst must not just give designers bulk HFE analysis data, but provide it in forms that are useful to the designer.

Designers will vary in their preferences for HFE analysis data presentation formats. Therefore, the designers should be asked for their preferences. It will help designers to decide their preferences if the analysis results are first synthesized into a briefing memorandum containing a readable narrative description of the user's tasks, why tasks are done, and what the user or user organization achieves from the tasks. This description can be supplemented with examples of the types of task description detail that are available in the task analysis data base. This memorandum will give designers a concrete basis for deciding what information formats would be useful, and for establishing some working procedures for getting HFE analysis data on request. Having the HFE analysis data available on some form of computer database makes it easy to provide HFE data in whatever form is needed. A database also makes it easy to extract information for selected user tasks or task variables.

Operations scenarios are another form of HFE data that can be useful to design. Detailed operations scenarios are not always available from system analysis information. At some stage of design, designers will (or should) want to review designs against detailed operations scenarios. These scenarios can be created from mission and task analysis information. Since operations scenarios will be needed for final design tests, it may be useful to provide them early in design so that they can serve both a design function and a testing function. In this connection, it is not unknown for design teams to learn only the week before a design test that several event sequences in operations were not anticipated in the system design. For example, a design might support a user task sequence, but not allow the user to suspend the sequence part way through and switch to another task - the type of thing that users have to do in real operations. Having scenarios available early in design will help to prevent this type of situation from happening.

4.3.2.4.1.2 Access to General HFE Information

Designers will have questions about HFE matters that can be answered either from the HFE specialist's personal knowledge, or from HFE literature. The questions will normally be generated by two types of specific design questions: 1) questions about the best way to execute a design element, and 2) questions about how users will perform if a particular design option is adopted. To provide useful answers to both types of questions, the HFE specialist must make sure that he or she understands the design problem, and the context of the problem. Answers should be precise and prescriptive, with no more qualifications than are absolutely necessary. The most useful form of an answer to a question of design is one that implicitly or explicitly provides the design solution: that is, a suggestion on how to execute the design in a way that would solve

the design problem. The most useful answer to a user performance question is either a quantitative estimate of likely performance, or a considered opinion of whether or not the design option will generate adequate user performance.

For other than questions about inconsequential matters, it is good practice to provide written answers to HFE questions. Writing answers ensures that answers are thought out. Keeping records of answers ensures consistency, and avoids misunderstandings. Once an answer has been passed along, it should be followed up to see if the answer was adequate, and if the design problem was solved. Follow-up provides an effective form of continuing HFE program quality assurance information, and helps to refine procedures for providing information.

HFE questions should never be answered by handing over copies of HFE technical articles, lists of references, or anything that leaves the person requesting the information having to do significant homework to get an answer. Providing literature and references is only acceptable when literature and references are specifically requested.

Finally, design processes generate questions at times of their own choosing, not at times convenient to the designer or the persons the designer may have to go to for answers. Therefore, HFE specialists need to be available to respond to requests for HFE assistance on short notice. Attempting to deal with *ad hoc* requests through scheduled liaison or review meetings is seldom effective. If the designer cannot get at least an initial response to a question when the question comes up, the designer will likely forego the benefit of HFE opinion, and press on with the design work.

4.3.2.5 Participation in Design Work

Occasional participation in design work is an effective way to maximize the efficiency of HFE design activities, and forestall HFE design problems. Two types of participation are useful: 1) participation in design planning, and 2) informal HFE reviews of planned or completed design executions.

Having some participation in design concept development, or design planning, is an effective way to get a preview of the coming design work, and thus, a preview of the HFE design issues. Having this preview will help in anticipating needs for HFE analysis data, and provide an early start on getting answers to HFE questions that are likely to emerge. This early participation may also make it possible to head off situations likely to create HFE design problems.

Participation in design planning sessions should be done in a way that does not intrude on design work. The best times to participate are when specific design work is being planned. This is when the design of individual system components is getting under way and when getting a preview of upcoming design work will be possible. Although forestalling problems is always useful, HFE problems that become apparent only to the HFE specialist during design sessions should be dealt with in another medium. Unless asked to participate actively, the role of the HFE specialist in

design planning sessions should be kept to gathering information useful for anticipating HFE near-term requirements for HFE inputs.

A periodic informal HFE design review with the individual or individuals responsible for designing a system component can accomplish three things: 1) identify designers' HFE information needs, 2) contribute HFE information to the design work, and 3) forestall the appearance of HFE inadequacies. Whenever possible, forestalling HFE design problems is much better than leaving them to show up in later formal the HFE design reviews specified in the CCIS HFE Requirements (see below Section 4.3.2.8, page 173). Ideally, the formal HFE design review will confirm the adequacy of the HFE features of the system design. If the formal review turns up significant problems, it will mean that the various measures taken to contribute HFE information to design either failed or were not adopted.

4.3.2.6 Identify and Apply HFE Design Standards and Guidelines

This section describes the application of HFE standards and guidelines to system design. The section is divided into four subsections:

1 a summary of the sections of the CCIS HFE Requirements associated with applying HFE standards and guidelines to system design;

2 a discussion of criteria to use in selecting standards and guidelines for a specific development project;

3 advice on setting up a system-specific human-computer interface style guide, and

4 a discussion of the roles that HFE specialists play in applying HFE standards and guidelines to system design.

4.3.2.6.1 CCIS HFE Requirements and HFE Standards and Guidelines

The CCIS HFE Requirements (5.2.2.1) requires that the contractor: 1) identify and adopt specific published HFE standards and guidelines as project-specific HFE design standards for the design of users' work spaces, and the design of human-computer interfaces, and 2) create a system-specific style guide to record the conventions used in the interface design. CCIS HFE Requirements (5.2.3) specifies the use of MIL-STD-1472D [12] for equipment, work station and work space design.

The CCIS HFE Requirements (6.4.2) specifies the use of MIL-STD-46855 [2] as a standard for conducting HFE work associated with purpose-built hardware. MIL-STD-46855 may be invoked in the contract if the CCIS to be developed contains hardware components requiring significant development effort. However, assuming that the CCIS will be predominantly a computer-based system, MIL-STD-46855's application will not be discussed here.

Reporting on plans for applying HFE standards and guidelines is specified in Paragraph 6.3.7.2 of the Human Factors Engineering Plan DID. The DID titled Human Factors Engineering

Simulation Plan specifies how to report the use of simulation methods and other devices for checking compliance with standards and guidelines in Paragraph 6.2.3. The DID, Human Factors Engineering Design Approach Document, describes how to report on, among other things, the application of HFE standards and guidelines in the system design. The DID, Human Factors Engineering Design Review Report, describes reporting on how standards and guidelines were used in HFE design reviews.

4.3.2.6.2 Choosing Standards and Guidelines

Standards and guidelines are required for two system components: 1) equipment, work space and work station design, and 2) human-computer interface design.

4.3.2.6.2.1 Standard for Equipment, Work Space and Work Station Design

The use of MIL-STD-1472D [12][26] as a standard for work space design is already required, so what standard to use for work space design is not an issue. However, not all Paragraphs of MIL-STD-1472D will apply to every system, so identifying which sections apply to the project will be necessary. Table 6 lists the main sections of MIL-STD-1472D. Sections 5.6 and 5.7 are the ones directly applicable to work space design. However, depending on the scope and nature of the CCIS being developed almost all of the topics except 5.14 have potential application.

Section 5.15 of MIL-STD-1472D also contains standards for human-computer interface design. Although the standards in Section 5.15 do not cover all, or even many interface design issues, they should be respected in the system interface design. The next section discusses interface design guidelines.

4.3.2.6.2.2 Human-computer Interface Design Guidelines

The production of human-computer interface design guidelines has been a substantial industry in the recent past. So, there are several sets of guidelines to choose from for application to a specific project. Since all interface guidelines documents contain good design advice based on HFE principles and empirical data, the technical soundness of the guidance given in a guidelines need not be a concern. However, guidelines documents do vary from one to another in other respects. Guidelines vary on two dimensions: 1) the scope of the guidelines, and 2) the accessibility of the design guidance.

The chosen guidelines should cover all the interface design issues that are likely to arise in the project. Table 7 shows a list of guidelines documents with brief descriptions of their contents.

---

[26] MIL-STD-1472 is periodically revised, so the reader should enquire to see if there is a version later than version D when the standard is adopted for a specific project.

Table 6 Descriptions of equipment and facilities design topics covered in MIL-STD-1472D.

Topic	Description
5.1 Control/display integration	Position, movement and gain relationships between control and display elements.
5.2 Visual displays	Design and arrangement of indicators, gauges, CRT displays and other information.
5.3 Audio displays	Design of audio warnings, speech transmission equipment, communications equipment controls.
5.4 Controls	Control design, movement, arrangement, location, operation with handwear.
5.5 Labelling	Labelling of equipment, location of labels, design of label characters.
5.6 Anthropometry	Anthropometric data for making equipment dimensions appropriate operator physical characteristics.
5.7 Workspace design	Workspace layout, console design and access.
5.8 Environment	Heating, lighting, noise and vibration limits for operating environments.
5.9 Design for maintainer	Mounting of equipment, access to components, test points and failure indications.
5.10 Design of equipment for remote handling	Equipment to be handled remotely and design of remote manipulation devices.
5.11 Small systems and equipment	Design for carrying and portability.
5.12 Operational and maintenance ground/shipboard vehicles	Design of operating controls, layouts and access for vehicles.
5.13 Hazards and safety	Safety labels, identification of hazardous equipment components and protection from electrical, mechanical and other hazards.
5.14 Aerospace vehicle compartments	Crew and passenger compartments, access and emergency evacuation.
5.15 User-computer interface	Design of displays, keyboards, pointing devices, system response times, user-computer interaction procedures.

Using the criterion of scope of coverage, it may or may not be difficult to identify a set of guidelines with contents that are congruent with the project's design issues. For example, the Goddard Space Center guidelines [15] provides the right coverage for a primarily text-based application that uses a windows interface. As a contrasting example, the ATCCS guidelines have generally good coverage of almost any interface design issue one might come across. The ATCCS guidelines also deal with map displays, something most guidelines do not do. However, the ATCCS guidelines covering older technology would not apply to windows interfaces.

Table 7 Human computer design guidelines suitable for application to CCIS development.

Interface Design Guidelines	Comments
Human engineering design criteria for military systems, equipment, and facilities (MIL-STD-1472D). [12] *	- Covers all equipment and facilities design and includes some interface design. - Covers only minimum interface design requirements. - Good on physical design requirements for I/O devices. - Lacks windows coverage.
OSF/Motif Style Guide [13] *	- Designed for windows applications. - Restricted to ensuring that software developed under OSF/Motif conforms to OSF/Motif conventions. - Does not cover many CUI design issues.
Human Factors Design Guidelines for the Army Tactical Command and Control System (ATCCS) Soldier-Machine Interface: Version 2.0 [14] *	- Designed for (Land Force) CCIS applications. - Very detailed and comprehensive (700+ pages). - Size and detail makes these guidelines inconvenient to use.
Human Computer Interface Style Guide, Version 2 [68]	- Based on Version 1 of ATCCS, but considerably shorter. - Material on maps, windows, decision aids and embedded help. - Literature authorities provided for individual guidelines. - Version 3 available.
User-Interface Guidelines (Goddard Space Flight Center) [15] *	- Covers windows interface design. - Concise and comparatively accessible. - Little coverage of graphics displays.
Department of Defense Handbook: Human Engineering Guidelines for Management Information Systems (MIL-HDBK-761A) [16] *	- Developed for tactical missile planning tasks. - Covers dialogue/interactive control, data entry, data display, job performance aids, expert systems, data communication, data protection. - Covers expert system design, but sketchily. - Covers design process including function and task analysis. - Covers graphical control techniques, but not Windows systems. - Good coverage of tactical map graphics.
The AWIS Common User/System Interface Style Guide [69]	- Supplements OSF/Motif Style Guide for U.S. Army WWMCCS Information System applications.
Macintosh™ interface design rules.	- Simple concise set of general principles for Macintosh application developers to follow in interface design. - Applicable windows and graphics design. - Lack detailed guidance for specific interface design problems.
User/Computer Interface (MIL-STD-1801 (USAF)) [70].	- Replaces interface design sections of MIL-STD-1472C for USAF. - Has same shortcomings as MIL-STD 1472D.
* Also listed in Section 11, page 4.	

The accessibility of the information in guidelines is important because most guidelines contain more than 100 pages of what amounts to minutiae. The bulkiest set of guidelines is the older ATCCS guidelines which runs to more than 700 pages. The prospect of sorting through many pages of detail to find out, for example, the best place to put a command button on a window display is bound to discourage designers and HFE specialists alike. Probably the best way around this problem is to get an electronic copy of the guidelines document that will permit at least string searches, and preferably hyper text links, for the locations of needed information. An electronic copy of a guidelines document can also be edited to remove guidelines material that will not be relevant to the design task, or to extract a collection of guidelines tailored to a specific design task. Being already familiar with a guidelines document also makes a document's contents more accessible than they would be otherwise.

In choosing a set of interface guidelines, and deciding how to use it, it needs to be kept in mind that interface guidelines share several significant limitations [71]. Interface guidelines typically give directions for designing individual interface elements, but leave it to the designer to combine these elements into an effective interface design. Consequently, an interface can come out with elements that represent good design, but with a combination of elements that leaves an ineffective design. Similarly, guidelines do not provide devices for checking if either the elements of an interface design, or the interface as a whole will produce effective user performance.

In addition, to giving the designer little guidance on how to combine design elements, the designer may have difficulty in translating the guidance on individual interface elements into an appropriate design execution. For example, de Souza and Bevan [72] in an experiment showed that designers had difficulty interpreting 90% of the general guidelines that were to be applied to a specific design task. Tetzlaff and Schwartz [73] found a similar result in another experiment.

In summary, while interface design guidelines provide sound HFE advice on detailed interface design, they also have three important limitations:

1 the information they contain is often difficult to find;

2 the guidance is predominantly about interface elements rather than on combining elements to produce a good interface design, and

3 the design guidance once found will not necessarily be correctly interpreted or applied by the designer.

These limitations suggest that just selecting an interface design guidelines document and handing it over to the design team is not likely going to generate an interface design with good HFE characteristics. A way of overcoming the limitations of published guidelines is to rely on a system-specific style document containing the interface design conventions to be used for the interface design. Doing this would consist of using a published guidelines document as a reference source, and then using the system-specific style document as the workaday reference for interface design. The system-specific style guide is discussed next.

4.3.2.6.3 System-specific Style Guide

A system-specific style guide is a set of interface design guidelines tailored to the interface being designed. Some form of a system-specific guide will be needed no matter what set of published guidelines is adopted. Published guidelines generally suggest more than one recommended way of designing a specific interface element, so for the sake of consistency, one of these ways must be settled on for the entire system. In addition, there will many conventions that must be standardized for the system: for example: time and date displays, abbreviations, user organization terminology conventions, and so on. All these conventions should be incorporated into the system style guide.

The HFE content of the style guide is a project HFE specialist's responsibility. For HFE purposes, the subject matter of the style guide is anything that a user will see on a display, and anything that a user has to do to operate the system. A convenient way to create a system style guide is to give it a structure parallel with whatever published guidelines are adopted for the system. Parts that concern universal interface elements can be cross-referenced to published guidelines. Other parts can be filled in as the need arises.

A system style guide will eventually represent something of a chicken-and-egg problem. Some of the guide's contents must be created before the corresponding design work is done. Other contents will be records of whatever style convention was adopted for an interface design element. Building up the style guide as design moves along avoids putting things into the guide that will never be referred to, and it avoids needless time-consuming speculation about what will be needed. However, for this approach to be workable, it must be kept in a pace with the emerging interface design. For example, the first time that a question about a style convention comes up and is decided, the adopted convention should be entered in the guide so that it will be available when the question comes up again. If design gets too far ahead of the style guide, designers will have to do otherwise needless design revisions to get the interface into compliance with the style guide.

The best way to make the style guide contents accessible is to put it in an electronic medium that is available to designers online. The precise medium and its functionality are a matter of the guide-users' preferences. A style guide can be a standalone document, or a section of a larger document containing all the system's design conventions. If the style document is a section of a larger document, the interface design conventions should be kept to this section and not scattered among other design conventions.

Designers should not be expected to work with a copy of the style guide constantly at hand, or even be expected to pay much attention to it during design work. The natural way to develop something like a screen or windows display is to draw it up and get it working, and then go back and revise it to conform to the style guide and any other conventions that need to be incorporated. Therefore, the most practical way to apply the style guide conventions to the interface design is to review each interface component against the guide when the initial design of the component is completed. A single window form and its input and output procedures represents a component of about the right size for reviewing. The review can be done either by HFE personnel or by design

team personnel. If standard design practice is being followed, someone other than the person who designed the component will check for compliance with the style guide. Whether this person is another member of the design team or a member of the HFE team does not matter.

4.3.2.6.4 The role of HFE Specialists in Applying Standards and Guidelines

The HFE specialist has four roles in identifying and applying HFE standards and guidelines:

1 recommend the standards and guidelines to use for the project;

2 help in the production and maintenance of a system-specific interface design style guide;

3 help in applying standards and guidelines to design, and

4 review design executions for compliance with the adopted HFE standards and guidelines.

These four roles represent basic requirements specified in the CCIS HFE Requirements. However, except for the fourth role, which will be dealt with in Section 4.3.2.8, (page 173), the CCIS HFE Requirements also gives leeway for assigning the extent and division of responsibility for filling each role.

Identifying and recommending HFE standards and guidelines to apply to design is a responsibility that the HFE specialist should be best qualified to do. Production and maintenance of the system-specific style guide requires the participation of both design personnel and HFE personnel. HFE participation in style guide production can be greater or lesser depending on the organization of the project. At a minimum, the HFE specialist should be responsible for the HFE principles and guidelines embodied in the style guide. This responsibility can be extended to production and maintenance of the style guide, including obtaining the design team input needed to define the design components to be covered in the style guide. Alternatively, the style guide can be a design team responsibility that includes getting HFE advice on the specific guidelines to be included in the guide.

HFE help in applying standards and guidelines to design should include at least responsibility for interpreting guidelines. The help can extend to giving advice on design problems where the proper application of a standard or guideline unclear, or where some compromise has to be made in applying a standard or guideline.

The preceding subsections described HFE activities that cut across more than one subsystem for which requirements are specified in the CCIS HFE Requirements. The following section reviews the HFE requirements for the design individual subsystems, and describes HFE activities associated with the design of a subsystem.

4.3.2.7 HFE Participation in Subsystem Design

The CCIS HFE Requirements specifies requirements for the detailed design of what will be called here for convenience, five subsystems: 1) human-computer interfaces; 2) equipment, work spaces

and work stations; 3) decision support applications; 4) automated functions; and 5) user manuals. The following sections describe the HFE requirements for these subsystems, and suggest how HFE specialists can help designers in ensuring that each subsystem will meet its HFE requirements. HFE design review responsibilities will not be covered. Design review is described later in Section 4.3.2.8 (page 173).

4.3.2.7.1 HFE in Human-computer Interface Design

The detailed design requirements for human-computer interface design are given in CCIS HFE Requirements (5.2.2), which begins by saying that human-computer interface design shall ensure design consistency and minimize the workload capacity that users must devote to human-computer dialogue tasks. The subparagraphs of 5.2.2 specify essentially two ways of getting design consistency and minimizing workload:

1 designing interfaces so that they comply with published guidelines on interface design, and

2 applying a user-centred approach to the interface design.

The guidelines requirement has already been covered in Section 4.3.2.6 (page 152), which describes how to select guidelines, produce a system specific style guide, and how HFE personnel can help in applying guidelines to interface design. The second requirement will be covered in the remainder of this section.

CCIS HFE Requirements (5.2.2.2) describes the user-centred design requirement. This requirement can be interpreted as aiming to achieve the five goals listed in Table 8.

Table 8 HFE goals of human-computer interface design.

1 The interface is tailored to the organization and sequence of tasks that the user is performing when using each computer application.
2 When the user is using the computer application to perform a task, information displays contain all the information needed for the task - and no more information than is needed for the task.
3 Required interface dialogue operations match the cognitive structure of the user's tasks: for example, the sequence of data input for a form corresponds to the way the data are ordered in the user's memory, or ordered in whatever medium is the user's source for the data.
4 The workload required by interface operating tasks is always well within the user's available workload capacity. This seems self-evident, but it is easily overlooked if the interface for a user's task does not take into consideration what else the user might be doing while using the interface, and what types of stress the user might be under while using the system.
5 The interface allows the user to acquire a mental model of each software application so that the mental model: 1) is a reasonable functional representation of how the software application or application object system is structured; 2) helps the user to remember how to do application operating tasks, and 3) allows the user to learn new operating tasks quickly, or work out how to do operating tasks that the user has never done before.

As stated, these goals represent somewhat subjective qualities. This is mainly because each represents a collection of both computer software and human characteristics. Broken down into their elements, they can be identified with specific human and computer characteristics that can be objectively defined and measured. Broken down however, the goals become less succinct and meaningful. Left in the form in which they are listed, they should provide an adequate basis for at least initial interface design - adequate both for the designer and the HFE specialist.

To serve any useful function in interface design, the HFE specialist has to have an understanding of the interface design approach being used. This is true whether the specialist's role is kept to providing HFE analysis data to the designers, or extends to taking an active advisory role in developing the interface design. Liu [74] gives a review of interface design methods. This review, and literature referenced in it, should be helpful for getting an understanding of available interface design approaches.

Interface design approaches vary from informal to formal. Informal methods rely on an essentially intuitive design process based on the designers' consultation with users. Strongly formal methods rely on abstract mathematical or quasi-mathematical models of the interactions between the user and the computer. The degree to which a formal method will have practical application in a specific CCIS project is difficult to predict. Suggestive, however, is the fact that examples of real applications of formal methods are noticeably sparse in the literature describing these methods. In practice, the actual process of interface design may be closer to the rough-and-ready approach.

From an HFE perspective, a design approach that will achieve the goals in Table 8 includes the following steps.

1 From the HFE analysis data, get an understanding of the user's tasks including:

- 1.1 the order in which tasks are done (or the order that should be most natural to the user),
- 1.2 what the user is trying to accomplish with the tasks,
- 1.3 the user's physical environment,
- 1.4 other tasks that could interrupt the user or that the user may have to keep track of,
- 1.5 operations scenario events that will trigger the user's task, influence how the task is performed, and define interface display default conditions, and
- 1.6 identify task constraints such as time and accuracy requirements, and needs for task verification.

2 Construct the ideas and output forms (windows) using the descriptions of the user's tasks to guide the design of the forms: including the division of functions into different forms; and on individual forms, the locations of labels, data input fields, data displays, user command buttons and menus. Current development tools make it easy to construct something approaching a functional prototype.

3 Incorporate conventions from the project's designated interface design guidelines and the system-specific style guide. If the design contains elements that are not covered in any

guidelines or in the style guide, choose a convention for the element, and update the style guide.

4 Do walkthroughs of the interface design using the task analysis information to decide if the interface fully supports the user's tasks, and in particular achieves the five design goals in Table 8. Revise the interface design as necessary.

At this point the design should be ready for an HFE design review (see Section 4.3.2.8, page 173). After any revisions made from the results of the interface design review, design can, if it has not already done so, continue on to developing a fully functional prototype ready for user testing.

The design procedure just outlined can be conducted as a typical informal design process, or as a component of a procedure based on a formal design model. In either case, the primary objective is to ensure that the interface achieves the HFE design goals in Table 8. The procedure can also be broken into segments organized according to the designers' divisions of work and responsibilities.

Although interface design guidelines and general HFE literature will provide direction on most interface design issues, a few items deserve extra attention in a command and control environment. These are described in the following paragraphs.

Pointing Devices

A wide range of pointing devices is available for computer input tasks. The characteristics and applications of different devices are described in [75]. One aspect of pointing devices that is not often mentioned in the literature is the design of software to encourage efficient use of the user's pointing device. Except tasks that require extended use of a pointing device, creating or manipulating graphics for example, a pointing device is not a very efficient input device where a simple keyboard command is, or could be available. It is particularly inefficient when the user is continually forced to switch between the keyboard and pointing device. Despite this, a lot of software forces the user to use the pointing device by not providing keystroke commands when keystroke commands could easily be provided. Lack of keyboard commands has a compounding effect because once the user starts using the pointer device during a task (because there is no keyboard command), he or she will likely keep on using the pointer to complete the task even though more efficient keyboard commands are available for completing the task. As this suggests, the HFE specialist should make a special effort to ensure that all interface components support keyboard commands.

Icons

Icons, used with a pointing device, provide non keyboard control actions [74]. Icon systems have been popular interface control devices for many microcomputer applications, although their use may be gradually going out of fashion. Many computer users, particularly new users, or users

who are not skilled with keyboards, seem to like icon interfaces. Designers may favour icons because they add graphic interest to the display. However, there is little evidence that icon-pointer systems yield much in the way of user efficiency benefits, especially for experienced users. Unlabelled icons whose meanings are obscure will slow performance, and may encourage inefficient usage patterns. An icon large enough to provide enough space for a meaningful graphic and a label, and provide an adequate pointing device target, also perforce takes up display space that might be put to better use. Small icons, sometimes called icon buttons, save space, but are difficult to make meaningful, and difficult to capture with a pointing device.

Unless a proposed icon is already familiar to the prospective users, it should be tested to see if users will correctly interpret its meaning. Campbell and Druce [76] provide a good example of how to conduct a user evaluation of the clarity, meaningfulness, confusion susceptibility, fitness of purpose, and memory retention for a set of candidate icons.

Usually, a labelled command button or text menu item is easier to comprehend than an icon. Labelled commands and menu items provide pointer targets that are just as easy to capture as icons are. All this suggests that icons yield few interface design benefits. In addition, icons require extra design effort and create extra loads on computer resources. In sum, icons should not be included in the interface unless a specific case can be made for using them.

Combat-level Terminal Devices

Nearly all information systems require data inputs that involve converting manual data into electronic data. For line operations data, it is tempting to place the point of data entry and conversion at the level of line operations where the data originate. As a result, a new information system can turn line operations personnel into data entry clerks by forcing these personnel to type data into a computer terminal instead of reporting data by voice or manually recording data at their own convenience.

In military operations, the line operations level is the combat level, and forcing data entry to this level can impose a considerable workload increase on combat personnel who have, at least in their view, more important things to do. Along with adding the data entry workload, an additional workload may come from having to use a less than adequate data entry device. Combat conditions, in land operations for example, may make it impossible to provide a data entry device that is as easy to use as a conventional microcomputer. To date, combat level requirements for portability and ruggedness have meant data terminal devices with displays and keyboards whose HFE characteristics would be considered far from adequate in any environment, let alone a combat environment.

Combat-level data entry should not be rejected out-of-hand. However, any requirements for what would amount to combat-level computer terminals should be considered very carefully before being adopted. If such a requirement is adopted, then the terminal devices should have to meet the same HFE standards that other computer devices have to meet. In addition, the workload

effects of data input in the field need to be carefully analysed to ensure that data input tasks will have no adverse impacts on the safety and effectiveness of combat personnel.

The need for a keyboard is a major barrier to having a practical combat level data device. A possible way around this would be a device based on speech control. Although existing speech recognition technology is probably not adequate to support a combat level device, the technology continues to improve rapidly [77]. Therefore, a speech-based device may very soon be practical.

Pixel Dimensions and Clarity of Computer Displays

Two points that deserve special attention in designing computer displays. One concerns the amount of text information that can be displayed on a display. The other concerns a common misconception about display size.

Displays with pixel dimensions in the order of 1000 pixels wide and 800 pixels high make it possible to get a large amount of legible text information onto a display - as much as 70 or so lines of text. This text capacity sometimes tempts programmers to create text displays, particularly data input forms (or windows), containing very high text information densities that are difficult for the user to comprehend and manipulate. One reason for this phenomenon is probably that in windows programming languages, putting all the user input fields for a function on one form is easier than creating two forms and dividing the input fields between forms. However, putting many input fields one form often produces a display that is difficult to read, or is even incomprehensible. Although displays with large pixel dimensions allow very small text characters to be legible, this legibility does not necessarily translate into displays that are readable. Usually, keeping text character sizes in text input and output forms close to the character sizes in older 20 to 25 line text displays will impose a design discipline that will produce more readable displays.

The misconception about display size concerns the fact that many users believe that the larger a display is the more clarity will it have. Although users have different ideas of what they mean by clarity, they commonly agree that bigger is better. As much as clarity depends on visual or image resolution, clarity essentially depends on the pixel density of the display for a *specified viewing distance*. Thus, if two displays have the same pixel dimensions, both will have the same clarity when each is viewed so that each subtends the same visual angle. In short, the absolute size of the display does not matter, it is viewing distance that matters. In turn, if a workstation's configuration determines the user's viewing distance, the display size should be chosen so that it will be appropriate for the viewing distance. It should not be the biggest display that the user can afford.

Visual clarity in the true sense depends on display resolution, which in CRT displays, is primarily a function of the pixel luminance profile [78]. Where there is a genuine need for high resolution displays, map graphics displays for example, the characteristics of the individual pixel are the first parameters that should be used to select a display. Once the pixel characteristics are decided, the display pixel dimensions can be decided from the work station viewing distance.

Material on computer displays and workstation design can be found in Grandjean [79], Smith and Cohen [80] and Engel and Townsend [17].

4.3.2.7.2 HFE in the Design of Equipment, Work Spaces and Work Stations

The CCIS HFE Requirements (5.2.3) describes the requirements for equipment, work space and work station design. Human Factors Engineering Plan DID describes how to report on plans for HFE participation in equipment and facilities design. Paragraph 6.4.5 of the DID, Human Factors Engineering Design Approach Document, describes the requirements for reporting HFE work in equipment and facilities design.

The HFE specialist's participation in physical facilities design will depend on the facilities designer's experience with military requirements. Contractor personnel experienced in military equipment and facilities design are usually already familiar with at least one version of MIL-STD-1472, and may need little direct assistance from HFE specialists in applying it to the design.

No matter to what degree the designer is already familiar with MIL-STD-1472D, the HFE specialist can help the designer by identifying the parts of MIL-STD-1472D that will be applicable to the design. The contents of Table 6 (page 155) should provide a starting point for identifying the applicable sections. The applicable items in MIL-STD-1472D can be formulated as a checklist that the designer can use to check designs against the MIL-STD-1472D standards. Since an HFE design review will be done later, this checklist can also be used as an equipment and facilities design review checklist (see Section 4.3.2.8, page 173).

The HFE specialist can also help with finding solutions to design problems. Section 4 (Control System Interface Design) of [17] contains a good deal of material useful for solving equipment and facilities design problems. Both human factors handbook editions [7], [8] contain information on equipment and facilities design.

Equipment and facilities design is usually more intuitive and unsystematic than software design. Design typically starts with rough drawings and proceeds through stages of increasingly detailed design. At the early stages, facilities layouts should be given the most attention. Layouts constrain interactions among personnel. The HFE analysis data will provide a basis for ensuring that facilities layouts make necessary interactions among personnel as convenient as possible, and do not create barriers to necessary face-to-face communications, or obstructions to views of important information displays. Personnel interactions can be studied by using link analysis techniques. An example of using link analysis for a command and control facility layout problem is given in Evans, Walker, and Beevis [81]. This study is also described in Beevis *et al.* [7]. As Beevis *et al.* also point out, link analysis and related facilities layout analyses can be performed using computer methods (e.g., Hendy [82]).

The detailed design stage is the stage at which most of the MIL-STD-1472D requirements will be applied to the design. This is also the stage at which individual work space and work station

designs begin to emerge. If the CCIS is fundamentally a conventional server system with conventional microcomputer terminals, much of the work space and work station design will be already determined by the computer equipment. If the CCIS includes purpose-built consoles, the work space and work station design will include questions of console configuration, display and control layout, and associated considerations. Section 4 of [17] includes material on these design questions, as do the two editions of the human factors handbooks.

Although the substance of equipment and facilities design is different from the substance of human-computer interface design, the roles that HFE specialists play in the two design activities are much the same: that is, provides designers with useful HFE analysis data, and contributes HFE knowledge to the system design.

4.3.2.7.3 HFE in Decision Support Application Design

HFE responsibilities in decision support applications design are specified in CCIS HFE Requirements (5.2.4). Paragraph 6.3.7.4 of the DID, Human Factors Engineering Plan, specifies how to report plans for HFE participation in decision support applications design. Paragraph 6.4.4 in the DID, Human Factors Engineering Design Approach Document, describes the requirements for reporting on the results of HFE activities in designing decision support applications. To the extent that decision support applications include human-computer interfaces, the CCIS HFE Requirements for human-computer interface design will also apply to decision support applications design (see Section 4.3.2.7.1, page 159).

The CCIS HFE Requirements specifications for HFE activities in decision support application design lists eight requirements. The following subsections describe these requirements.

4.3.2.7.3.1 Role of the User's Mental Model

CCIS HFE Requirements (5.2.4.a) specifies that *the application's architecture and input and output facilities provide the user with an efficient mental model of the application's functions, and is congruent with the cognitive structure of the decision problem*. An application that supports an efficient and accurate mental model of the application's operation achieves two complementary objectives: 1) it minimizes the chances of user errors, and 2) maximizes the ease with which the user can learn and correctly apply the application to solving a problem. Although no one quite knows what a mental model is, it is a notion that almost everyone intuitively understands. The notion is widely used in HFE to understand how people comprehend and predict system behaviour. Background on mental model applications can be found in Norman [4], Rasmussen, Pejtersen, and Goodstein [83], Wickens and Carswell [52], Moray [84], and in almost any general HFE textbook.

A mental model is a mental representation of a system that allows the user to understand the structure and behaviour of the system, and to predict the system's behaviour as a function of user actions. Mental models are usually physical representations or physical metaphors of the system.

Mental models of complex systems typically represent only the major and most important system functions. Thus, mental models are usually incomplete representations of the system.

Mental models are acquired through experience with the system - experience that may include formal training. Skilled system users have more accurate and more elaborate mental models than novice users do. A skilled user's mental model includes the most critical and most frequently used of a system's functions. A novice's mental model may include unimportant and seldom used functions along with important and frequently used functions. In short, skilled users have more efficient mental models than novices do.

There is no established formula estimating the extent to which a software application, or any other system, lends itself to helping the user to form an effective and efficient mental model of a system. However, the nature of mental models suggests the following five criteria that can be used to judge whether a system will promote the development of an effective mental model.

1 *The primary user-application interface is a physical analogy with the software's architecture and functions.* Conventional word processing and spreadsheet software are examples of applications whose interfaces provide natural mental models. The word processor interface is a literal representation of what the user is trying to produce, and it behaves essentially the way putting pen to paper would. Similarly, spreadsheet software interfaces are literal representations of the paper spreadsheet table a person would use to organize and solve the types of problems that involve tabular data. It is significant that the inspiration for the first spreadsheet software came from realizing that a computer display could be constructed to look like a paper spreadsheet, and include imbedded computing functions that would do what would normally be done by hand or on a calculator. It may be a characteristic of a good application designer is the ability to recognize intuitively the value of supporting users' natural mental models.

2 *Where the application does not naturally suggest a mental model, the application interface provides devices that the user can employ to develop an effective mental model of the application.* Database software provides examples of applications whose functions can be difficult to unravel, particularly when the database structure gets much beyond a simple table structure. With a complex database, the user is often provided with few tools for visualizing the structure of the database, or for understanding what the application does when the user makes inquiries, adds records, or generates reports. Expert system software often presents the same types of problems. Process control application design is probably the best source of design solutions that support mental models for applications that do not naturally evoke effective mental models. The mimic diagram showing a graphical representation of the system is an example of a ready-made mental model that shows a system's structure, functions and behaviour [17]. The mimic diagram is an example of so-called configurational and ecological displays used in a variety of process control applications. Many of these display approaches can be used in decision support applications to support the user's mental model of the application (see Rasmussen, *et al.* [83], and Moray [84]).

3 *The application supports mental models of the computation and decision problems that the application is used to solve.* An application can support a mental model of its own operation without necessarily supporting a mental model of the computations that the application is

designed to perform. A spreadsheet program is an example of an application that provides a natural mental model for what it does, along with a natural mental model for most of the computation problems that spreadsheets are used to solve. However, spreadsheets can also be used for any non tabular computations that the program's formulae can be used to calculate. For a non tabular computation, the user often has to spend more time setting up the program to show the steps in the computations, than the time spent setting up the actual computation. Here, the user has to provide his or her own support for a mental model showing the steps in the computation. A CAD (Computer-aided Design) program is an example of programs that provide devices to support, say, the user's three-dimensional conception of a drawing object when the drawing interface is two-dimensional. The device that would best support a mental model of the problem depends, of course on the problem. The main thing is to recognize that the user will naturally try generating a mental model, and that the application design should ensure that the user's model is efficient and effective.

4 *The effectiveness of the scheme chosen for inducing or supporting a mental model has been empirically verified.* For a given application, some mental models can be more effective than others, and what serves as an effective mental model for one type of user may not be effective for another type of user. Fault diagnosis behaviour is an area where the use of mental models has been intensively studied (Bond [85]). Research in this area shows that what makes or supports an effective mental model depends on variables such as the expertise of the user, and the application metaphor used to support the mental model. Tilley [86], for example, showed that fault diagnosis is more effective with mimic diagrams organised as a tree than when the mimics are organized to represent the layout of the system. These results imply that the effectiveness of any mental model scheme proposed for an application should be verified either by empirical test, or from past research findings.

5 *The application training aids and user manuals promote the development of an efficient mental model of the application's operation.* Training aids and user manuals can promote mental model development by explaining how the application works. Diagrams of the application's structure and operation will be especially helpful. Also helpful are worked examples of applications of the different types of problems the program is intended to solve.

The degree to which an application meets these criteria is to a considerable degree a matter of judgment. Nevertheless, the criteria cover the primary variables that govern whether or not an application supports the development of an effective mental model of an application's operation.

4.3.2.7.3.2 Display and Confirmation of Input Data

The CCIS HFE Requirements (5.2.4.b) specifies that *all data and parameters used by the application for a specific decision problem are displayed to the user, and in the case of critical data and parameters, confirmed by the user before computations are performed.* This is an obvious requirement, but if decisions made from the application have potentially critical impacts on command and control operations, it is wise to check carefully that the requirement is met. In applications that do computations on data from multiple input forms, and possibly multiple stored data sources, inadvertently programming the application so that it does its computations without the user being able to see the input parameters all at once before computing starts is easy to do.

Having a single display that shows the user all the important input parameters ensures as much as possible that the computed output corresponds to the user's complete understanding of the input conditions. If the program has no single display of input conditions, the user must rely on memory for some of, or all these conditions. Having to rely on memory for checking input conditions makes the user vulnerable to errors in setting up input conditions: for example, forgetting to change a parameter for a new problem, or forgetting that a parameter was changed. This will result in the user thinking that a problem solution came from one set of input conditions when the solution really came from a different set of input conditions.

4.3.2.7.3.3 Traceability and Verification of Decision Aid Outputs

The CCIS HFE Requirements (5.2.4.c and d) contains two related requirements: 1) that *the application provides the user with facilities for tracing how the application arrived at a specific problem solution by, for example, displaying intermediate results or displaying logic traces of problem solutions,* and 2) that *the user can independently verify the accuracy and plausibility of problem solutions.* These requirements apply particularly to any type of expert system decision aid.

If an expert system user cannot trace how the application arrived at a problem solution, the user has no way of verifying the solution, and consequently, no basis for trusting the accuracy or utility of the solution [87] [84]. Users will not use decision aids that they cannot trust. Even if users are willing to use the aid, the aid's problem solving algorithm could still have faults that the user will have no way of detecting. Finally, if users are required to use a decision aid whose solutions cannot be traced, users will then be implicitly freed of responsibility for the consequences of decisions made from the aid's solutions.

The requirement for independent verification of the accuracy and plausibility of problem solutions is essentially a general requirement for having redundancy in the decision making system. Traceability is one way of providing a means for independent verification. Other ways include building plausibility checks into the decision aid application, and creating standard operating procedures to ensure that more than one user verifies a problem solution. Whatever, measures are used to check decision aid solutions, they must minimize the chances that a single user and the decision aid application can combine their functions in a way that produces a serious decision making error.

4.3.2.7.3.4 Training and Job Aids for Decision Support

The CCIS HFE Requirements (5.2.4.e) specifies that *training, user manuals, on-line help, and other job aids are adequate to ensure that the user will understand the application's logic and computational algorithms, risks of error, problem conditions required for using the application, and any limitations that could lead to the user making errors in command and control decision making tasks.* This is another measure intended to prevent decision aid misuses leading to decision making errors.

A good deal of the documentation of early versions of general purpose decision support software, spreadsheet programs for example, included educational material on how to use the software to solve the various problems for which the software was meant. The trend in more recent versions of commercial software is to limit documentation to that needed to operate the software, and to leave information about practical applications to third-party publications. To ensure that this documentation trend does not creep into the CCIS project's environment, all necessary steps must be taken to ensure that the application training materials and documentation explain the logic of the application, and the application's limitations.

4.3.2.7.3.5 User Qualifications for Using Decision Aids

The CCIS HFE Requirements (5.2.4.f) specifies that *users will have the education, military training and experience required to make safe and effective use of the application.* This requirement deals with several issues related to user qualifications. They are

1 the correspondence between users' expertise in the subject matter of the decision aid and the expertise imbedded in the decision aid, and

2 provision of training to ensure that users correctly exploit the decision aid's capabilities in support of command and control decision making.

Decision aids, particularly expert system types of aids are often intended to provide an expert's capabilities to non experts [87]. In command and control, this is a questionable approach. Ultimately, commanders and staff are responsible for the decisions they make. This implies that an expert system application in a command and control environment must not embody more expertise than the application users themselves possess. A related issue is the degree to which the subject matter experts who contributed knowledge to the application's development typified the intended users, and were familiar with user organization's decision making procedures and doctrine. That is, the decision aid must accurately reflect the organization's decision making procedures and doctrine.

Training to exploit a decision aid is important because having a decision aid available will not necessarily lead to it being used effectively. Demonstrating this point, Geiselman *et al.* [88] in testing a computer aid for artillery fire planning found that different planning teams used different approaches to working out a fire plan, and as it happened, produced solutions of varying merit depending on the approach used. Individual differences in planning naval operations were similarly found in a test of a command and control performance measurement system called HEAT (Headquarters Effectiveness Assessment Tool) [89]. In this test [90], different planning teams had different approaches to solving tactical manoeuvre problems. These and other examples in the literature suggest that when it comes to complex high level tasks, different command and control personnel do not necessarily do every task the same way. Similarly they may not be uniformly capable of properly exploiting a decision aid: that is, unless they are trained to use the aid properly.

4.3.2.7.3.6 Workload Impacts of Decision Support Aids

The CCIS HFE Requirements (5.2.4.g) specifies that *the workload demand of using the application: (1) is consistent with the benefits of using the application, (2) is less than the workload demands of alternative methods of dealing with the decision problem, and (3) does not reduce the workload capacity the user must keep available for other tasks*. Unless a decision aid provides an outstanding improvement in the quality of decision making, this requirement could be the most important consideration in whether or not a decision aid is effective.

Usually, the time taken to complete a decision making task while using the decision aid compared with the time taken to complete the task by an alternative means will provide an adequate measure of the decision aid's impact on user workload. Alternative means will usually be the means employed in the predecessor command and control system. Therefore, the predecessor system task analysis data can probably provide the baseline workload measures.

4.3.2.7.3.7 Decision Support Aid Risk Assessment

The CCIS HFE Requirements does not include an explicit provision for assessing the risks connected with using a decision support aid. When the decision support aid is merely a labour-saving device that automates low-level routine tasks, the risks inherent in using the aid are probably small. When the decision aid automates complex problem solving and decision making functions in support of critical command and control tasks, the risks for making serious command and control decision errors may become greater than without automation.

Automation of cognitive tasks is a growing trend in nearly all fields of providing support for human activities. HFE research in aviation and nuclear power plant management suggests that while it is increasingly possible to automate complex cognitive tasks, there is a tendency to automate the things that are easiest to automate (and also often easiest for humans to do), and leave the difficult cognitive tasks to the human (see for example Wiener [91], Moray [84], and Sheridan [92]). This creates a situation where the automatic system can go happily along managing on its own until some condition exceeds the automatic system's permissions. When this happens, the automatic system may suddenly hand things over to a human who has little idea of what has been going on, or of what led to matters getting outside the automatic system's control capabilities. The human is now in the awkward and possibly dangerous situation of having to respond to an emergency without having any advance warning of things going wrong. In contrast, if there were no automatic system, the human would likely have advance knowledge of the problem, and either be able to deal with the problem, or be able to prevent the problem altogether. Thus, while automation has taken over some human workload, and possibly reduced human errors on the now automated tasks, it may also have increased the chances of more serious human errors in the more complex non automated tasks.

The lessons from recent experience suggest that the development of advanced decision aids must include a thorough assessment of the risks associated with removing the human from the control loop, and then asking the human to take care of difficult situations that the automated system

cannot handle. Right now, there is no systematic way to evaluate these risks. However, literature like that just cited suggests that the design of the automated components of the system must include an intensive analysis of the events or situations that the automated system probably cannot handle, and a complementary analysis of how to ensure that the human component of the system will be kept prepared to handle situations that are outside the automated component's capabilities.

Automation in command and control decision tasks may be less advanced than in other fields. On the other hand, the nature of command and control decision making is more complex than in other fields, and there is little theory available for developing automated support (Foster [93]). Current decision aiding in command and control eschews the early conceptions of fully automated systems capable of generating fully worked out battle plan alternatives [94]. Current approaches concentrate on giving commanders and staff clearer and more comprehensive pictures of tactical situations, rather than on automating actual decision making (Andriole [95]). This orientation is probably just what is needed to avoid the risks inherent in systems that have humans only for dealing with awkward situations that the automation cannot cope with. While this orientation persists, command and control decision support systems should avoid the risks that have emerged in other fields.

4.3.2.7.4 HFE in Automated Functions Design

The CCIS HFE Requirements (5.2.5) describes the requirements for automated functions design. Paragraph 6.3.7.4 of the DID, Human Factors Engineering Plan, specifies how to report plans for HFE participation in automated functions (and decision support applications) design. Paragraph 6.4.4 in the DID, Human Factors Engineering Design Approach Document, describes the requirements for reporting on the results of HFE activities in the design of automated functions.

For the purposes of the CCIS HFE Requirements, an automated function is any automation of manual tasks that is not classified as a decision support function. Examples of automation in this category include automation of essentially routine or clerical tasks that lend themselves to automation. The requirements for automated function design are similar to those for decision support application design. The essential requirements are the following three.

1 Users always have complete control over the automated function, and always know its status.

2 The automated function generates significant user workload savings compared with the manual alternative, is cost-effective, and produces significant user performance benefits.

3 Training and job aids ensure that the user understands how the function works, and minimize the chances of user errors resulting from misuse of the function or misinterpretation of the function's output.

These requirements are meant largely to prevent any unintended and unwanted effects that automation can have on user performance. Many of these unintended effects are described in Sarter, Woods and Billings [96].The most common unintended effects are

1 user workload not reduced as intended, but redistributed or increased,

2 changed or added attentional demands, and new knowledge demands,

3 failures in awareness of system status or system outputs,

4 new demands for coordination between personnel and system functions, and

5 additional training demands.

Since these five effects are the most commonly found adverse effects of automation, they can be used as criteria for determining if an automated system design is likely to have automation-specific adverse impacts on user performance. The HFE requirements for all other aspects of automated system design can be dealt with from the requirements for compliance with HFE standards and guidelines, and requirements for human-computer interface design.

4.3.2.7.5 HFE in the Production of User Manuals and Training

The CCIS HFE Requirements specifies that HFE principles are to be applied to the development of user manuals and training manuals to ensure completeness, clarity, technical accuracy, suitable format of information presentation, appropriate reading level, and appropriate technical sophistication. For purposes of software development, a user manual includes online help facilities, and any other devices intended to help users in learning and using the system. The force with which the CCIS HFE Requirements specification for user manuals should be invoked in system development will depend on the stage of design. At early stages of design, it will not be worth developing a complete set of user documentation because the design will probably change before any users get access to the system.

For an initial design, the user documentation should include at least a basic manual describing how to do the system design's operating tasks. If the design is a computer-based prototype, the documentation should be provided as an online help facility. Although there may be reluctance to provide a help function early in design, writing the help file is an effective way to detect gaps, inconsistencies and errors in the software functions. System operating task descriptions provide a good source for creating the help file documentation. Conversely, help files can be useful for getting descriptions of operating tasks.

If the design is going to be tested on representative users in a design test, the design should include, besides online help and system usage documentation, a training program adequate for the purposes of the design test. It may turn out that to get the design test running, the HFE team will have to be responsible for developing this training. A complete set of usage documentation will be needed as part of the final system design, and the final system design test.

The HFE specialist can contribute to the development of user manuals and training by giving the documentation developer knowledge about the users' capabilities and likely information needs. In this role, the HFE specialist can combine knowledge about the users' operations with HFE knowledge to provide advice on the requirements for documentation, and on the suitability of proposed documentation. The HFE standards and interface design guidelines literature provide basic advice on the construction of online help facilities. However, knowledge of users'

requirements will likely represent the HFE specialist's most useful contribution to the development of user manuals and related support materials.

4.3.2.8 HFE in Design Review

This section describes HFE activities in reviewing the system design to determine if the design meets HFE requirements. A design review is essentially a systematic inspection of the design to see if the design complies with system-specific HFE design requirements and general HFE design principles. A thorough design review is an efficient way to see if a design meets specific and general HFE requirements. However, though a positive design review outcome is very encouraging, it is no guarantee that the design will meet its ultimate user performance goals. Only the design testing methods described later in Section 4.4 (page 184) can show whether or not the design will meet performance goals.

The following description of design review methods begins with a description of the CCIS HFE Requirements specifications for design review activities. The remainder of the section describes the review procedures.

4.3.2.8.1 CCIS HFE Requirements Specifications for Design Review

The CCIS HFE Requirements (5.2.2 to 5.2.6) gives design review specifications for each of the five design activities: human-computer interface design; equipment, work station and work space design; decision support applications design; automated functions design, and the design of user manuals.

Paragraph 6.3.7 of the DID, Human Factors Engineering Plan, gives directions on reporting plans for HFE design reviews. Directions on reporting the results of design reviews are given in the DID titled Human Factors Engineering Design Review Report.

4.3.2.8.2 HFE Design Review Procedure

Although the technical substance of each design review will vary, the same basic procedure can be used to do each review. The most effective way to conduct a design review is to review the design against a checklist made up of items representing the design's HFE requirements. These requirements include both system-specific design requirements and requirements for compliance with HFE standards and guidelines. There are four main sources of items for a design review checklist:

1 user function and task analysis data describing the command and control tasks by which users are to achieve the system's design goals and design requirements;

2 descriptions of the tasks that users will perform to operate the design system;

3 measures of user performance and effectiveness defined for the system design, and

4 HFE standards and guidelines that the system is to comply with.

The first two sources implicitly include system design requirements. They divide users' tasks into 1) tasks that users want to get done by using the system, and 2) tasks that the user has to do to operate the system. Using task analysis data as sources for design review variables is supported by Malone and Micocci [97] who found that descriptions of users' tasks are one of the best sources for checklist items for design evaluation. In short, the essential purpose of a design review is to assess how well the design supports users' tasks, and is easy to operate.

The third source, previously defined measures of performance and effectiveness, can provide inspiration for checklist items, even if the measures cannot be applied in the review. In this connection, the design checklist can also serve to preview the usefulness of prospective design test measures. For example, including items that represent possible design test measures in the design review can suggest what would happen if the measures are used in a real design test. Results from items used only for previewing how performance and effectiveness measures might work later can be treated as a type of measurement pilot testing, and need not be reported as design review results.

Items in the fourth category, compliance with HFE standards and guidelines, can be drawn almost ready-made from the standards and guidelines documents used for the design.

Checklists have several potential limitations [98], [99]. For example, individual items that are too broad or too general are difficult to apply consistently, and do not yield prescriptive design advice. Similarly, items that reflect general design principles rather than specific design requirements do not produce prescriptive design advice. Most of these types of limitations can be avoided by constructing each checklist item so that it meets the requirements listed in Table 9.

Table 9 Requirements for design review checklist item construction.

1	The design quality to be assessed by the item should be directly observable.
2	The criterion for a design feature's acceptability should be a dichotomous specific quality or quantity that can be directly compared with the design feature being reviewed.
3	The checklist item should directly reflect a predefined system or user performance requirement, or a specific design standard or guideline.
4	The checklist item should provide an implicit recommendation for design improvement should the design fail to meet the item's acceptance criterion.
5	Checklist items should be organized to reflect logically related collections of design requirements or subsystems.

Items that meet the first two requirements in Table 9 ensure that the design review will be objective and produce results that can be independently verified. The second requirement also filters out potential checklist items that might require conducting tests, or making elaborate measurements to decide if a design feature is adequate. The third requirement, to have predefined acceptance criteria, ensures that the design review will be directly related to the system design requirements. It also ensures that the review's scope is consistent with the scope of the CCIS HFE Requirements. The fourth requirement, that the checklist item provide an

implicit design improvement recommendation, will usually be redundant if the checklist item meets the first three requirements. Nevertheless, keeping this requirement in mind will be a reminder that to be useful, the design review results must yield prescriptive design advice.

The fifth requirement concerns the organization and collective properties of the checklist items. Organizing the items into logical groups will help to make the review procedure efficient, and help to make the review results comprehensible.

Achieving a checklist in which every item meets the requirements in Table 9 is difficult to do. For example, items concerning user workloads will not be easily quantifiable. Checking a design against these types of items will demand professional judgment from the HFE specialist. When this is so, the specialist can provide an informed judgment during the review, and should a design element be judged inadequate, can provide a prescriptive recommendation for correcting the design. If a prospective checklist item does not lend itself to informed judgment, and an informed recommendation for a design correction, there is little point to putting the item on the checklist.

Once the checklist has been drafted, it should be reviewed with the designers to ensure that the checklist is complete and that the design acceptability criteria accurately represent the design requirements. Reviewing the checklist with the designers also ensures that the review results will not give the designers any undue surprises.

Checklist review results can and should be reported as recommendations for design corrections. Although giving any constructive advice that will produce an improved design is always useful, the design review report should not stray much beyond what can be derived directly from the checklist data. For example, summary evaluations of the overall merits of a design or design component are beyond what can be expected from checklist data. Checklist items are normally mutually independent, so aggregating results from separate items is not possible without making questionable item weighting assumptions. The sole purpose of a design review is to ensure that, to the degree possible, the design conforms with specific design standards and guidelines. The design review methodology seldom warrants drawing summary conclusions about the design as a whole.

Returning to the fifth requirement in Table 9, about logical item groupings, grouping checklist items according to topics in the CCIS HFE Requirements design review requirements would suit most situations. These topics are

1 compliance with the HFE standards and guidelines adopted for the system;

2 degree to which human-machine function allocations are consistent with design requirements and relative human-machine capabilities;

3 user interface's compliance with interface design principles;

4 user workload impacts;

5 assessment of system effectiveness;

6 user skill and knowledge requirements;

7 user training requirements, and

8 provisions for user manuals and training.

Besides being logical, these groupings also make it convenient to divide the design review into sessions, each session covering one or more of the listed groupings. Dividing the review into sessions may also allow more efficient scheduling of personnel time. For example, if participation is needed from design personnel for some groups of items but not others, a session with designers devoted just to the items needing designers' participation would avoid taking up needless time from designers.

The following subsections describe the types of checklist items that could be included in the different groupings, and suggested procedures for applying the items in a review.

4.3.2.8.2.1 Compliance with HFE Standards and Guidelines

Compliance with standards and guidelines can be determined in either of two ways: 1) by using a checklist made up from the standards and guidelines documents that were used for the design, or 2) by referring to the design documentation to confirm that standards and guidelines were complied with. If the design process included a systematic procedure to confirm compliance with HFE standards and guidelines, repeating the procedure is unnecessary.

Compliance with the standards and guidelines can be checked by doing walkthroughs of system functions, or for equipment and facilities, checking design dimensions against the standards.

4.3.2.8.2.2 System Function Allocations

The primary purpose of this review is to see if functions allocated to users are consistent with the principles of human-machine function allocations. A checklist for this review can be made up from the task analysis data and asking for each major task whether or not:

1 the functions allocated to users are consistent with the functions allocated to users earlier in the design process,

2 the users' functions are ones that humans in general and users in particular can do effectively,

3 any functions allocated to users that would be better allocated to hardware or software.

To keep this component of the review manageable, the tasks chosen for the review should be tasks that are performed frequently, and critical tasks in which quick and accurate performance is important for command and control effectiveness. The tasks in the review should include all user tasks, not just ones involving use of the CCIS. Unless all user tasks are considered in the review, there will be no way of deciding whether or not users have been left to do tasks that could have been automated. Tasks that contain significant routine repetitive work are the ones most likely to reflect improper function allocations.

4.3.2.8.2.3 User Interface Review

This review covers interface design components that would not have been covered in the review for compliance with HFE standards and guidelines (Section 4.3.2.8.2.1, page 176); that is, assuming that a set of user interface design guidelines was used for the system's user interface design. Rather than being concerned with compliance with guidelines, this review is concerned with whether or not the interface gives the user a convenient and efficient flow of work. Table 8, which lists the HFE goals of human-computer interface design (Section 4.3.2.7.1, page 159), provides a basis for a checklist for the interface design review. Depending on the degree of direct HFE participation in the interface design, the interface design review may be already complete.

4.3.2.8.2.4 User Workload Impacts

A good system design will yield an optimum workload demand that permits efficient, comfortable and error-free performance. Although there is no way to estimate workload demands imposed by a system accurately, design characteristics that cause undesirable levels workload are easy to identify. These are listed in Table 10, and can be used as the basis for a checklist to assess the workload characteristics of a design.

The first column in Table 10 lists factors that represent or create workload demand. The second column lists corresponding system design features that create unnecessary workload demand. A system design that has many of the features listed in the second column will impose unnecessary workload demands on the user.

Table 10 Task factors and design features that create workload demand.

Factors That Create Workload	Design Features That Create Unnecessary Workload
Searching for information from the environment	Complex displays that make it difficult for the user to find information needed for a task
Organizing and interpreting information	Requiring the user to organize and interpret information that the system could organize and interpret
Remembering facts and rules to make decisions	Requiring the user to rely on memory for facts and rules when the system could provide access to this information
Organizing and performing actions in response to events	Requiring the user to perform unnecessarily complicated sequences of actions to perform tasks
Tasks with serious consequences from error	Failure to provide protections from such thing as accidentally erasing files
High precision control movement tasks	Unnecessary pointing tasks, and pointing devices with inadequate precision or inappropriate control-display gain
Lack of ability or skill for performing tasks	Requiring users to have extraordinary abilities or extensive training to perform system operating tasks

In addition to reviewing the design against the criteria in Table 10, the design should be reviewed in the context of all the users' tasks. This ensures that the user always has the workload capacity available to do not just system operating tasks, but also other tasks that the user may have to look after simultaneously. This can be done by looking at task analysis data to identify tasks that must be done in parallel with system operating tasks. Alternatively, one or more of the function analysis methods (Section 4.2.4, page 49) or simulation methods (Section 4.2.6, page 103) can be used to identify workload problems.

Depending on the system, other workload analyses are possible. For example, when operational sequence data are available, task analysis information can be combined with the operational sequence data to create computer simulations of operations. These simulations can show if a design would allow users to handle the workloads expected in real operations, or show if a particular design would improve users' workload capacities compared with their capacities in the existing system.

Additional workload assessment approaches can be found in Tsang and Wilson [51], and in Beevis *et al.* [18] in a section titled Performance Prediction. Some methods described by Tsang and Wilson, and Beevis *et al.* have already been described here as function analysis and simulation methods. Others described in Tsang and Wilson, and in Beevis *et al.* are workload measurement methods suitable for application in a design test. Although the methods are not directly applicable to design reviews, they may provide ideas for design review checklist items. For example, many subjective workload measurement instruments can be easily converted to checklist items that the HFE specialist can use to assess a design's workload characteristics.

A precise estimate of a design's workload demands is possible only in a design test (see Section 4.4, page 184). In a design review, the workload assessment can do no more than identify design characteristics likely to create unnecessary workload demands, or identify design features and user tasks whose workload characteristics deserve more careful analysis.

4.3.2.8.2.5 Assessment of System Effectiveness

System effectiveness is the extent to which the system supports the user's command and control performance goals. If measures of command and control effectiveness have been identified according to the CCIS HFE Requirements (5.1.2), as part of the HFE analyses, these measures can be used as checklist variables for reviewing the effectiveness of the system design. Since system effectiveness is often difficult to relate to specific design features, extra care needs to be taken to be sure that system effectiveness checklist items meet the checklist item requirements given in Table 9 (page 174). Assessing system effectiveness with variables that cannot provide objective and prescriptive advice for design improvements is of no value. As emphasized at the beginning of this section, only the design testing methods described later in Section 4.4 (page 184) can confirm whether a design will be effective.

An assessment of system effectiveness can be an occasion not only for applying system effectiveness measures, but also an opportunity for identifying new measures, or improving on

ones already identified. Since a design review involves a systematic review of the design in light of users' tasks, the review provides a good medium for identifying effectiveness measures that may turn out to be useful in a design test. Similarly, effectiveness variables that cannot be assessed in a design review should be marked out for use in the design testing.

4.3.2.8.2.6 Skill and Knowledge Requirements

Skill and knowledge requirements here mean the skills and knowledge that users will be expected to have *before* they start learning to operate the system. Information about the skill and knowledge capabilities of the user population should be provided by the acquisition authority. The design's skill and knowledge requirements do not include the user training requirements. They include only the background skills and knowledge that a user would need for learning how to operate the system. Skill and knowledge requirements can be assessed with answers to the following questions.

1 Have the education, training, and experience of the intended users been specified by the acquisition authority?

2 Are the skills and knowledge needed to operate the system consistent with the education, training and capabilities of the users?

3 Is the mix of skill and knowledge requirements for each individual user consistent with the mix of skill and knowledge capabilities possessed by each individual user?

4 Is the mix of skill and knowledge requirements consistent with the mix of skill and knowledge capabilities present in the existing command and control organization?

5 Will the skill and knowledge requirements have adverse impacts on personnel requirements by

  5.1 requiring personnel with capabilities not available in the current user organization, or

  5.2 failing to reduce personnel skill and knowledge requirements as specified in the design requirements?

6 Are there any design changes that would reduce skill and knowledge requirements, or reduce personnel requirements?

Depending on the system being reviewed, these questions can be asked of the system as a whole, or repeated for individual users, individual system functions, or individual subsystems. Providing useful advice on design improvements should be the principle for deciding how to divide the system design into components for assessing skill and knowledge requirements.

4.3.2.8.2.7 Provisions for User Manuals and Training

Provisions for user training include online help facilities, user manuals, job aids such as quick reference cards, and classroom training that users will need both to learn to operate the system, and to maintain their operating skills and knowledge. Ultimately, the user training program will

be derived from the user function and task descriptions to identify skills and knowledge requirements needed to operate the system. Even if the training requirements have not been developed by the time the design is reviewed, getting an outline of the training requirements from the function and task analysis data should be possible.

The content of a design review of provisions for job aids and training will depend on the stage of system design. As discussed earlier in Section 4.3.2.7.5 (page 172), the completeness of user documentation and training will depend on the stage of system development. For an initial design, the user documentation should include at least a basic manual describing how to perform the system design's operating tasks, and for software prototypes, an online help facility. If the design is going to be tested in a design test, documentation should include a user training curriculum. For a final design, there should be a complete set of user manuals and training materials.

The design review for user manuals and training can be conducted by reviewing the materials against the task descriptions of the design's operating tasks and identifying instances of:

1 operating procedures for which there is no user manual coverage, online help, or training support;

2 inconsistencies between user documentation and operating procedures;

3 inadequacies in the readability or comprehensibility of documentation;

4 system functions or design features that will create inordinate training demands;

5 documentation that has technical errors concerning the user organization's operations, and

6 any significant infringements of HFE design principles.

Again, as mentioned earlier in Section 4.3.2.7.5 (page 172), the HFE specialist's most useful contribution in this review is being able to take the user's point of view to identify gaps in the user documentation and identify ways to improve the documentation. Although not necessarily part of the review, a review of documentation and training can sometimes reveal potential design improvements: for example, simpler ways of performing operating procedures.

If the system design has a predecessor system, the review of user manuals and training is an opportune time to see if the new system and the old system operating procedures have unnecessary inconsistencies. To minimize negative transfer effects, a new system's operation should be as similar as possible to the operation of the predecessor system. This will reduce training demands and eliminate operating errors caused by absentmindedly applying operating habits from the old system to the new system where they are inappropriate.

Finally, the review of provisions for user manuals and training should contain comment on whether or not the provisions as a whole are consistent with the stage of system design. During development, producing manuals and training may be thought of as a far off future activity that will be looked after at implementation. Instead manuals and training are integral components of the system, and their development must be considered at all stages of system design.

4.3.2.8.2.8 Standard Operating Procedures

There are two types of standard operating procedures:

1 standard operating procedures for operating the system itself, and

2 other standard operating procedures that may be indirectly affected by the system.

The system design should include standard system operating procedures in its user documentation wherever there is more than one way of performing system operations. The standard procedures should be whichever procedures offer the safest, most effective and most efficient use of the system. Provisions for standard operating procedures can be reviewed as part of the review of provisions for user manuals and training. The review should identify any instances where standard operating procedures are needed but not specified. Where standard operating procedures are specified, the review should assess the HFE merits of the procedures.

The design review should identify any standard operating procedures that are not part of the system, but may be affected by the system. An example of such a procedure is one that users cannot perform as effectively as previously because the users have new tasks connected with using the new system. The example could also be a procedure that users can do more effectively because the new system will give them more time to do the procedure. Indirect impacts on users' operating procedures are all part of how well the new system will integrate into the total command and control operation. When to review thoroughly how well a system will integrate into operations depends on the nature of the system. However, it should not be left until design is too far advanced to permit design changes that would improve the system's integration into operations.

The user function and task descriptions are the obvious basis for determining how well a system will integrate into operations. Reviewing these descriptions will show which functions or tasks are likely to be affected by the system, and whether the effects are likely to be positive, neutral or adverse. Where the impacts are adverse, the remedies are either to alter the system design or to change the affected operating procedures.

4.3.2.8.3 Scheduling HFE Design Reviews

An HFE design review can be conducted whenever design is advanced enough to review; the review would yield useful design information, and design is not so advanced that changing the design would be difficult if the review showed that change was needed. Design reviews should be scheduled so that they can serve as design feedback in the manner shown in a Figure 24 (which is reproduced from the CCIS HFE Requirements User Manual).

In Figure 24, the HFE requirements analyses and other HFE design activities provide the initial HFE input to the design process. The control loop takes the design output through three HFE activities: 1) updating the HFE analyses describing users' functions and tasks in the design; 2) from the results of these analyses, doing the HFE design reviews, and 3) from the results of the design reviews, identifying HFE design improvements to feed back into the design process. The

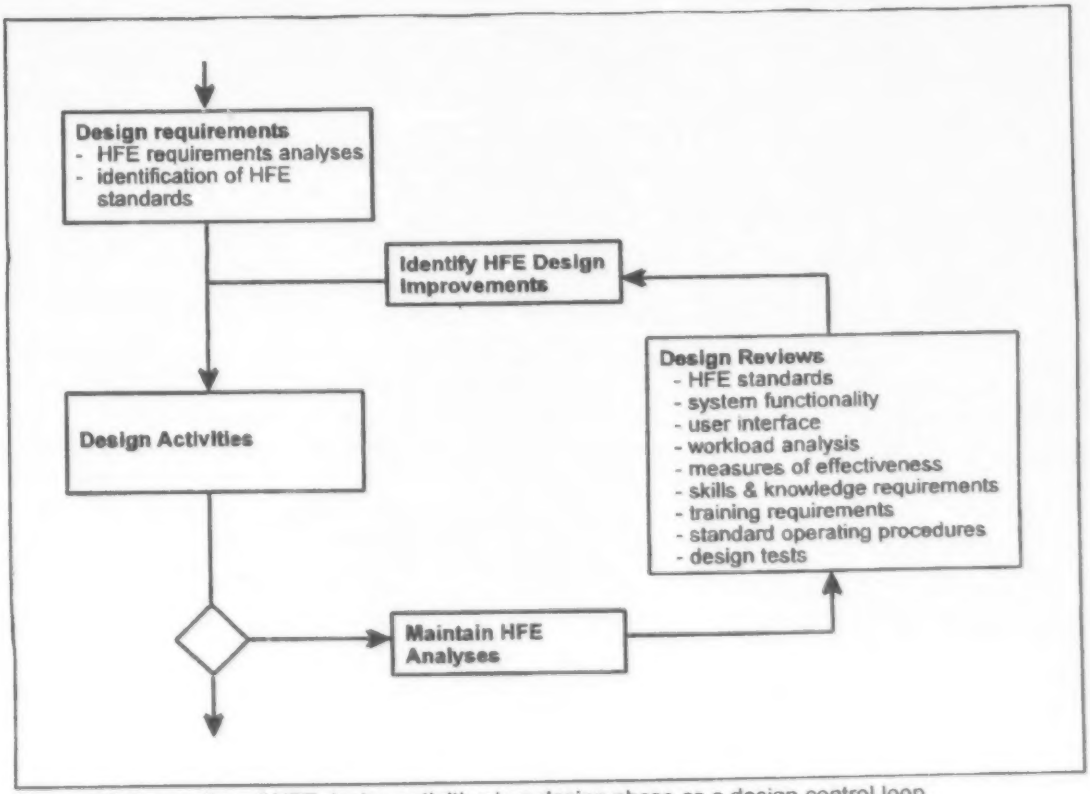

Figure 24 Conception of HFE design activities in a design phase as a design control loop.

loop can be repeated until the design meets the HFE requirements. The timing of the loop should aim at making the most efficient use of designers' and HFE specialists' time.

4.3.2.8.4 Summary of HFE Design Review

The following summarizes HFE activities in design review.

1 The purpose of a design review is to assess a design's compliance with the CCIS HFE Requirements before design testing.

2 The design review can be conducted using a checklist made up of items drawn from four sources:

2.1 descriptions of user functions and tasks supported by the system,

2.2 descriptions of the tasks that users do to operate the system,

2.3 previously identified measures of performance and effectiveness, and

2.4 HFE standards and guidelines applicable to the design.

3 Each design review checklist item should meet four criteria:

3.1 refers to a design feature that is directly observable;

3.2 includes a criterion of acceptability that is a specific quality or quantity that can be directly compared with the design feature being reviewed;

3.3 directly reflect a predefined design requirement, or a specific design standard or guideline, and

3.4 provide an implicit recommendation for design improvement should the design fail to meet the item's acceptance criterion.

4 The design review checklist can be divided into sections corresponding to the different sets of activities that make up HFE activities in design. These are

4.1 compliance with the HFE standards and guidelines adopted for the system;

4.2 degree to which human-machine function allocations are consistent with design requirements and relative human-machine capabilities;

4.3 user interface's compliance with interface design principles;

4.4 user workload impacts;

4.5 assessment of system effectiveness;

4.6 user skill and knowledge requirements;

4.7 user training requirements, and

4.8 provisions for user manuals and training.

5 Reporting on the results of design reviews should concentrate on providing prescriptive advice on how to correct whatever design shortcomings were found in the review.

The HFE design review completes HFE activities in system design. The next set of HFE activities consists of conducting design tests. Design testing activities will be described in Section 4.4, immediately following the next section which summarizes HFE activities in design.

4.3.2.9 Summary of HFE Activities in Design

The HFE activities in design break into two sets corresponding to two phases of system design: 1) identification of design requirements, and 2) detailed design. The activities corresponding to these two are the following:

1 *design requirements activities* including:

1.1 identifying user support requirements,

1.2 identifying decision support requirements,

1.3 participating in function allocation,

1.4 doing tests to verify decision support and function allocation proposals, and

1.5 participating in iterative development activities.

2 *detailed design activities* including:

2.1 keeping the HFE analyses current with the system design,

2.2 making HFE data and information available to the design effort,

2.3 identifying and assisting in the application of HFE standards and guidelines to design,

2.4 contributing to human-computer interface design, equipment and facilities design, decision support applications design, and automated functions design, and

2.5 conducting HFE design reviews to assess compliance with the HFE standards and guidelines, human-machine function allocations, compliance with user interface design requirements, user workload impacts, assessment of system effectiveness, user skill and knowledge requirements, user training requirements, and user manuals and training.

4.4 HFE Testing Methods

An HFE test is any experiment that includes gathering data from human subjects to test a hypothesis about the performance of the subjects. In system development, HFE tests are normally conducted for one of four possible reasons: 1) to demonstrate or confirm that a subsystem design meets user performance requirements; 2) to provide a final system acceptance trial at the end of development; 3) to provide predecessor system baseline data for comparison with the new system's performance, and 4) to provide an empirical answer to an HFE question that cannot be answered from the HFE literature. Although tests are conducted for different purposes, the basic procedure for conducting a test is generally the same no matter what the purpose of the test. In the following, HFE testing procedures will be described as design testing procedures, but on the understanding that the same procedures would apply to tests conducted for any other purpose.

The remainder of this section gives advice on doing HFE tests. The first subsection will describe the CCIS HFE Requirements specifications for design testing. The next subsection describes a concrete example of a design test. The remaining subsections describe the tasks that must be done to conduct a design test. The tasks are

1 create the HFE test plan;

2 specify the test measures to be used in the test;

3 construct instruments for administering the test measures;

4 create a test medium or a design representation for testing the design;

5 define the test's experimental design;

6 construct the operations scenarios for testing the design;

7 develop test participant training;

8 pilot test the test administration procedure;

9 select and recruit test participants;

10 arrange for conducting the test, and

11 analyse and report test results.

4.4.1 CCIS HFE Requirements Specifications for HFE Tests

The CCIS HFE Requirements (5.3) gives the requirements for design tests. The central requirements are that:

1 The designs of any subsystems operated by users or support users' tasks are to be tested for compliance with HFE requirements, and

2 The complete integrated system design is to be tested to:

  2.1 show that the system as a whole meets HFE design standards and system-specific HFE requirements;

  2.2 show that the system will support users' command and control effectiveness requirements;

  2.3 provide quantitative measures of system performance that depend on the users' interactions with the system, and

  2.4 show that the system does not have any undesirable design features, or create undesirable procedural demands.

3 Tests are to be conducted on design representations that provide realistic renderings of the eventual system, and are to be based on test scenarios that provide realistic renderings of the user organization's operating scenarios.

Nearly all the requirements in Paragraph 5.3 reinforce or elaborate these three basic requirements.

Requirements for HFE tests are also described in CCIS HFE Requirements (5.1.5) in connection with the HFE analysis requirements. Here, testing is required where user capabilities associated with decision support requirements and function allocations cannot be verified from human factors engineering literature or data bases. Overall, the CCIS HFE Requirements requires empirical testing when the HFE literature is not adequate to support important design decisions.

Requirements for reporting test plans are given in two DIDs: 1) the Human Factors Engineering Plan, and 2) the Human Factors Engineering Test Plan. Test plans reported in the Human Factors Engineering Plan should be outline plans showing how testing will fit in with HFE analysis and design activities. The Human Factors Engineering Test Plan is intended for reporting the detailed plan for each test. The requirements for reporting test plans aim at ensuring that proposed tests are justified and that their methodologies are sound.

The DID, Human Factors Engineering Simulation Plan, describes requirements for reporting plans for constructing any design representations that will be needed for testing. Overall, the Human

Factors Engineering Simulation Plan should be reserved for design representations that will be used in more than one test, or for design representations whose cost will be substantial. If a design representation is going to be used in only one test, and is not particularly expensive, the plan for building it can be included in the Human Factors Engineering Test Plan.

The DID, Human Factors Engineering Test Report, describes the requirements for reporting on HFE tests. These requirements are essentially the conventional standards for reporting on any scientific experiment. These standards include the requirements: 1) that the report provides a basis for judging the validity of the results and conclusions, and 2) that the report contains enough information to enable the reader to repeat the experiment. Test results can also be reported in any DID where test results are pertinent to the subject of the report. However, every test report should have a separate companion report describing the experiment from which the test report's results were obtained.

4.4.2 An Example of A Design Test

This section briefly describes a concrete example of a design test. The purpose of the example is to provide a frame of reference for the subsequent sections describing the preparation and conduct of a design test. The example is not intended as a model of how a test should be conducted. It is intended only to provide a feel for the structure of a design test and the types of activities that go into doing a design test. The example is taken from a user test of an experimental CCIS intended for use in a battalion command post [55]. Some data from this study were used earlier to illustrate some of the HFE analysis methods (see, for example, Figure 4, page 44 and Figure 20, page 110).

The system that was tested consisted of a computer-based CCIS designed to mediate communications between the battalion command post and its higher brigade headquarters, and between the battalion command post and its subordinate subunits. The system was also designed to manage the command post's combat information including maintenance of the battle map. The test was designed to compare impacts of different design configurations on the information handling characteristics of the command post. There were three basic design configurations. In addition, each of two of the basic configurations had two further configuration variations.

Three teams of test participants acted as battalion command post staffs. Each team consisted of an operations officer, three duty officers, an operations clerk, and a radio operator. The participants were given two weeks of training before the actual test was conducted. This training included learning how to operate the different system configurations, and training on command post operating procedures related to use of the test system. This amount of training time might be thought excessive compared with the six days required to run the actual test. However, it was no more than required to learn how to operate a complex advanced system design.

Test exercises consisted of simulated combat operations sequences in which each participant team operated a command post in response to message traffic from higher brigade headquarters and from four lower (subordinate) subunits. Exercise controllers simulated voice message inputs to

the command post from brigade headquarters and subunits. The controllers worked from detailed scripts so that precisely worded messages were sent at predetermined times. For computer data links, the computer system automatically generated messages at predetermined times.

The exercise consisted of three battle sequences, each simulating the inputs for a battle group command post. Each sequence contained a total of 12 hours of operations. Each sequence was divided into three segments of four hours each. Each segment served as input for four hours of continuous operations for a participant team operating one design configuration.

The test experimental design (called a balanced Latin Square) is shown in Table 11. For the purposes of the test experimental design, a four-hour operations segment was called an iteration. In the table, the letters A, B, and C represent participant teams. The designations BG1, BG2 and BG3 represent battle groups. The numeric suffixes on the battle group designations represent battle sequence iterations. The identifiers 1, 2A, 2B, 3A, and 3B represent the different system design configurations. As the table shows, the test exercises were administered over a period of six days.

Table 11 Experimental design for CCIS design test example.

Day	Battle Groups		
1	A BG1.1 1	B BG2.1 2A	C BG1.1 3B
2	A BG1.2 2A	B BG2.2 3B	C BG1.2 1
3	A BG1.3 3A	B BG2.3 1	C BG1.3 2B
4	C BG1.1 3A	A BG2.1 1	B BG1.1 2B
5	C BG1.2 2A	A BG2.2 3B	B BG1.2 1
6	C BG1.3 1	A BG2.3 2B	B BG1.3 3A

Three battle sequences permitted the three teams to operate simultaneously, each on one battle sequence iteration. Each team ran six iterations. The last three iterations were a repetition of the first three iterations. Teams were switched among test configurations after each battle sequence iteration so that no team played the same battle sequence with the same design configuration more than once. All three battle sequences covered the same general operation. Thus, the three battle groups were operational elements in a common operation. This meant that participants switching from one battle sequence to the next could maintain some degree of continuity of perspective on the operation as a whole. The orders in which the teams operated the different configurations on different iterations were arranged to counterbalance sequential effects from learning and from interaction effects among design configurations.

Each iteration began with an operational order from the brigade headquarters. Iterations were designed to contain approximately equal numbers of reports of enemy contacts, subunit location reports, and the like. None of the sequences required the participants to make tactical decisions, and no participant acted as a unit commander. When a matter was to be referred to the unit commander, a member of the test administration team acted as a proxy for the commander.

The test measures included the following.

1 *A running record of the times at which messages were originated and processed through the command post.* These times were collected from the message logs. A set of linking codes were used to record the routing of messages that could be conceived of as a single message originating at, for example, a subunit and being sent through the command post to the brigade.

2 *The results of briefings given by duty officers to test administrators acting as liaison officers visiting the command post.* The requests for briefings occurred approximately every 30 minutes. Briefing responses were recorded on a specially formatted sequence of battle events. The sequence of events was pre coded so that the CP visitor could check off the events that the duty officer would be expected to mention in a briefing. The duty officer was expected to mention only tactically significant events.

3 *Duty officers' responses to a tactical situation knowledge questionnaire administered every two hours.* The questionnaire contained items covering knowledge of the tactical situation of each subunit, knowledge about enemy activity on the unit's boundaries, the unit's vulnerability to the enemy, and knowledge of orders currently in effect. Correct answers for the items on the tactical questionnaire were identified by reference to the sequence of events. Items on the questionnaire were designed to reflect factual knowledge about the tactical situation rather than judgments about, for example, what the enemy would do next.

4 *Responses to a questionnaire about the training that participants received for the test.* Each questionnaire item covered a particular system operation. Each participant rated the degree to which the operation was understood, and how important it was to understand the operation.

5 *Responses to an opinion questionnaire about the merits of various system features.* The questionnaire covered three topics: 1) the effectiveness of design system compared with existing manual systems; 2) opinions about specific design issues that had emerged during system development, and, 3) questions about the system's ergonomic features.

6 *Results of a user interface review.* The review was conducted by running through one of the test battle sequences, and observing the various system displays and required system usage actions as they came up during the battle sequence. Since the system's operation was undocumented, the review was restricted to observing obvious deviations from HFE design practice.

These test measures provided a mix of objective user performance measures and subjective opinion measures. Because the test was an *ad hoc* test, and there was no system operating task analysis information, the test measures were limited to ones that could be devised from command post standard operating procedures.

Since this design test example is only intended to illustrate the nature of a design test, most of the test results are of no particular interest. However, one result is worth discussing as an experimental design object lesson. The experimental design did not include a command post configuration representing the then existing manual command post operation. The user organization asserted that a manual configuration was unnecessary because, 'we already know how the manual system works.' As it turned out, 'we' did not know how the manual system

worked. Although the test results reliably showed differences and similarities in command and control user performance achieved with the different configurations, lack of comparison data for the manual operation left moot the question of whether or not any of the design system configurations yielded any better performance than the existing manual operation. This was particularly evident in the results from the tactical situation knowledge questionnaire.

Results from the tactical situation knowledge questionnaire showed that, on average, duty officers were only able to answer about 50 percent of the questionnaire items correctly. A score of 50 percent on knowledge of subunits' tactical situations seemed uncomfortably low. However, without a manual command post benchmark, it was impossible to tell if this result could be considered normal, below normal, or even an improvement on the manual system. Interpretation of the results for all the user performance measures suffered from this problem. The object lesson here is that if one has no predecessor system benchmarks, and no prior user performance requirements to be met by the new system, there is no way of drawing any conclusions about the merits of the new system. Thus, for example, it matters little if a new system shows first rate HFE characteristics if it cannot be shown that the system produces better user performance than the predecessor system did.

The design test just outlined is, as already mentioned, hardly a model study. However, it should give a general idea of the scope and nature of a design test, and particularly an idea of the work that has to go into planning and conducting a design test. The example should also show that the effort needed to produce valid, reliable and comprehensive test measures will always be modest compared with the logistics costs of running a design test. The effort to produce a sound design test will usually also be modest compared with the costs and benefits associated with the decisions to be made from the results of the design test. In short, stinting on the test development activities described in the following sections is not a cost-effective way to manage HFE resources.

4.4.3 HFE Test Plan

An HFE test plan is a plan for an individual HFE test: for example, a plan for a test of how well a proposed subsystem design meets user performance requirements, or a plan for a final system acceptance trial. A well-thought out detailed test plan is probably the most important factor contributing to achieving a successful test. A detailed plan is important for two reasons.

1 A detailed plan helps to ensure that all the factors that could influence the success of the test have been considered, and that the test will be technically sound.

2 Detailed planning is necessary for the coordination of the resources needed for the test, including:

  2.1 arranging for test participants,

  2.2 obtaining access to a system representation such as a system prototype to use as a test medium,

  2.3 constructing operations scenarios for use in the test,

2.4 getting the equipment and facilities needed for the test, and

2.5 coordinating the test activities with other system development activities.

The first reason is the universal reason for detailed planning in any experiment. That is, good planning is indispensable for efficient research and for producing reliable results. The second reason for detailed planning is specific to HFE testing in a system development environment. As the list of items under the second reason should suggest, many things that have to be arranged to run an HFE test, and consequently many things can go wrong. Good detailed planning is the only defence against things going wrong.

Detailed planning is also important because a design test usually requires much organizational coordination. Unlike a laboratory experimenter who has a good deal of autonomy and control over setting up an experiment, the HFE specialist running a design test must depend on many different people for getting the resources and cooperation needed to set up a test. These people include

1 *the PM and PMO staff* who are responsible for getting test participants from the user organization;

2 *the user organization* authorities who are responsible for making test participants available, and, depending on the test requirements, making equipment and facilities available;

3 *the test participants* themselves whose cooperation will be essential for the success of the test;

4 *subject matter experts* whose help will be needed to develop the operations scenarios to be used in the test;

5 *internal project management personnel* who are responsible for coordinating test activities with other system development activities, and for making equipment and facilities available for the test, and

6 *other project specialists* whose cooperation will be needed for such things as the production of, or access to the design representation to be used in the test, assistance with participant training, assistance with data collection during the test, and so on.

If the test is to run smoothly, all of these people will need access to the test plan well before the test. They will also need detailed direction for filling their roles and responsibilities in the test.

Detailed scheduling is important not only for coordinating the resources needed for the test, but also for coordinating test activities with other system development activities. Design tests will usually cause some interruption in design work, particularly any design work that cannot continue until the test results are complete. Scheduling test activities well in advance can help prevent interruptions in development activities by allowing managers to schedule other design work to replace any work halted for the design test.

The easiest way to work out a test plan is to use the Human Factors Engineering Test Plan DID as an outline plan. The DID covers all the issues discussed in the preceding paragraphs. As such it serves as a checklist for confirming that a test plan is complete, and that it covers all the

coordination matters that need to be included in the plan. Since it is almost inevitable that the test plan scheduling will change between the time when the plan is created and the time when the plan is executed, it is wise to make a list of all the people who will be involved in the test, and simultaneously, set up a procedure for keeping everyone informed about any changes in the test schedule. Although the PM is normally responsible for coordinating user organization participation in a test, the PM still has to rely on the contractor for keeping up to date on the status of the test plan.

The following subsections will describe how to do the different tasks that make up an HFE test. The tasks generally correspond to items specified in the Human Factors Engineering Test Plan DID.

4.4.4 Specification of Test Measures

Test measures are the measures of user performance that will be used to evaluate a proposed system design. Specifying test measures can be treated as a process of first identifying all the possible test measures that could be used in the test, and then selecting out the measures that will be most suitable for the design test. For various reasons that will be described in the following subsections, not all of the candidate measures will be suited or applicable to the design test. Arriving at a set of measures to be used in the design test can be done in three steps.

1 Identify the candidate measures from the HFE analyses and select a preliminary set that are applicable to the design test.

2 Select a subset of the candidate measures based on the likely adequacy of the measurement properties of the candidate measures.

3 Review the measures selected so far for completeness; add measures if necessary, and get rid of measures that are redundant.

The following subsections describe these steps.

4.4.4.1 Identifying Candidate Test Measures

The identification of candidate test measures will be described in three parts. The first deals with basic measures that are explicitly specified in the CCIS HFE Requirements. The second deals with user workload measures. The third discusses consulting with the users of the design test results to ensure that the candidate measures cover all the results users' information needs.

4.4.4.1.1 Test Measures Specified in the CCIS HFE Requirements

Primary candidate test measures are measures of user performance and command and control effectiveness as defined in the CCIS HFE Requirements (5.1.2). These measures, and how to identify them, were described earlier in Section 4.2.7 (page 121). By the time an HFE test is

being designed, these measures of performance and effectiveness should have been already identified in the relevant CCIS HFE Requirements data items: for example, the Human Factors Engineering Analysis Report. If the measures have not already been identified, the procedure described in Section 4.2.7 can be carried out to identify them.

Repeated here for convenience, the performance and effectiveness measures specified in the CCIS HFE Requirements are:

1 times required to execute command and control cycles for representative mission scenarios;

2 times taken to perform critical functions and tasks;

3 times to perform tasks on the command and control cycle critical path, and delays between critical path tasks;

4 times to access information required for critical tasks;

5 accuracy of performance of critical tasks;

6 accuracy of information produced by or required for critical tasks;

7 times required to perform tasks, and

8 measures of users' awareness and comprehension of tactical situation and tactical events.

Many of the items in this list are measures that can be applied to individual user tasks. They therefore represent potentially many individual measures corresponding to all the different tasks that users might do in a mission scenario. Usually, the possible measures will be more than would be practical to apply in a test. This means that the job of identifying test measures is a job of selecting a set that will be practical, and simultaneously, pertinent to the purpose of the test.

Four criteria can be used to filter out useful measures. The measures must

1 be valid for the test's purpose, or the system design requirements being addressed in the test,

2 represent performance critical to mission achievement;

3 be specifically influenced by the system design features that will be evaluated in the test, and

4 include criteria representing adequate performance or adequate effectiveness.

These criteria implicitly include the CCIS HFE Requirements specifications for measures of performance and effectiveness derived from HFE analysis data. In addition, they include criteria relating candidate measures to the purpose of the test and the system design features that are to be tested.

If a primary set of measures has not already been identified, Table 12, reproduced from [17] with minor editorial changes, provides a comprehensive checklist procedure for identifying candidate test measures for a design test. Table 12 contains four groups of questions that can be asked about a task or task sequence to decide if a test measure is needed, and to identify possible test measures. The first three groups of questions deal with user performance measures. These measures correspond to the measures of user performance and command and control effectiveness specified by the CCIS HFE Requirements as part of the HFE analysis requirements. Thus the

Table 12 Steps to select design test measures.

1 Review the user task descriptions to identify tasks that should be measured:
- does the task have to be performed within a required time?
- is there a required maximum error rate allowed for the task?
- is the task critical to the system mission?
- is the task likely to be prone to errors?
- what impact will errors have on the system mission?
- will errors affect the performance of other tasks?
- is the task performed frequently; would improved efficiency reduce overall user workload?
- is the task on the critical path of an important task sequence?
- are there critical events or conditions that the user must quickly recognize?
- does the task require the user to solve problems, make judgments, or divide attention between several displays?

2. Establish performance criteria:
- is there a critical, or desirable time limit for performing the task?
- is there a critical, or desirable maximum error rate?
- are there precision of performance standards that have to be maintained by the user?
- are there minimum reaction times that the user has to meet?
- are there qualitative characteristics of performance that reflect adequate performance?

3. Identify system design characteristics that influence user performance:
- what are the display features that are critical for the user's performance?
- what are the control features that are critical for the user's performance?
- what are the operating procedures that are critical for performance?
- are there any specific design modifications that might improve performance?

4. Identify the types of information that users could supply to explain performance:
- did the user feel adequately trained?
- did the user understand system operation adequately?
- did the task seem natural to perform, or was it awkward?
- did the user have to do a lot of thinking or remembering to perform the task correctly?
- did the user notice features that hampered performance?
- can the user think of design improvements that would improve performance?

items in Table 12 can be used to check that the HFE analysis measures are complete and comprehensive.
The fourth group of checklist items in Table 12 deal with questions that could be asked in questionnaires to get information from test participants' that would amplify or clarify the performance measurement results. These items are not required by the CCIS HFE Requirements, but they could provide useful information nevertheless. In particular, answers to these questions provide guidance on how to modify design features that were inadequate according to the design test results.

All the test measures suggested so far are related to command and control effectiveness. However, they are largely measures of user performance that support command and control effectiveness rather than being direct measure of command and control performance. Examples of more direct measures of command and control performance are listed in Table 13.

The first column of the table lists various time and accuracy measures. The second column lists command and control effectiveness examples corresponding to the different time and accuracy measures. Table 13 is reproduced from [19]. [19] contains a review of command and control performance measurement literature, along with a set of performance measures designed for land force command and control organizations. Although concentrated on land force operations, the

Table 13 Example of performance measures with command and control examples.

Performance Measure	Command and Control Example
Speed/Time Measures	
Time taken to perform a fixed task, or fixed sequence of tasks	Most planning tasks
Time taken to perform a variable task	Developing or selecting options or courses of action
Time taken to recognize or respond to an event	Time to recognize or respond to a critical enemy contact
Time taken to achieve a target state	Time to achieve a tactical objective
Percentage of time on target	Percentage of time logistics data bases are up-to-date
Number of events in a queue waiting to be processed	Number of messages awaiting action in a command post
Timeliness of responses	Maintenance of fire plan schedule; maintaining reporting schedules
Accuracy Measures	
Accuracy or precision of performance of tasks	Accuracy and completeness of information plotted on maps, and information maintained in data bases
Sensitivity for detecting system events	Recognition of events likely to require change in plan
Probability of making an error.	Errors in fire plan target schedules
Time taken to recognize that an error has been made	Time taken to recognize need for a plan alteration
Time taken to recover from an error	Time to alter a tactical plan segment
Knowledge of current system status	Comprehension of battle situation
Quality of decision	Quality of tactical plan

material can be adapted to any command and control operations. Although direct measures of command and control effectiveness are often attractive, their results may be difficult to link to specific features of the design being tested. For example, the quality of a tactical plan depends on many factors besides the system design being tested. If a command and control effectiveness measure cannot be linked to a system design feature, the measure will be of no use.

4.4.4.1.2 Identification of Workload Measures

Workload measures are given separate treatment here because workload considerations are given high priority throughout the CCIS HFE Requirements. No system design is acceptable unless the

workload imposed by it is within the workload capacity of the user. Similarly, minimizing user workload is always a primary virtue of a design.

Since there are no direct measures of workload, workload has to be measured by inference from other variables. Workloads can be inferred in two ways. One is by inference from the values of the time and accuracy of performance variables. The other is by questionnaire measures of workloads.

Workload demand and workload capacity are functions of a user's speed and accuracy of performance. Therefore, many speed and accuracy measures identified as candidate test measures can also be used, either individually or in combinations, to infer workload quantities. For example, the point at which a task demand becomes high enough to cause performance errors can usually be interpreted as the point at which the user's workload capacity has been exceeded. As this implies, candidate speed or accuracy measures can be sorted to identify ones that represent workload measures. These workload measures can be individual measures flagged as workload measures, or groups of measures that are mathematically combined to produce derived measures of workloads.

Questionnaire measures of workloads are called workload scales. A workload scale usually contains a set of items asking the respondent questions related to the amount of workload pressure experienced while doing a task. Several workload scales are described in Tsang and Wilson [51], Beevis *et al.* [18], and American National Standards Institute (ANSI) [100] which includes workload measures along with a useful compilation of other performance measures.

Among workload measurement scales, the Cooper-Harper scale is a particularly useful one[27]. The Cooper-Harper Scale is described in Wickens [101], which also contains a stylized version of the scale. The Cooper-Harper scale is a well-established scale of workload that was originally created to measure the workload impacts of aircraft control systems. The scale is based on a decision tree conception of a user's judgments of the difficulty of doing a task. The scale is ten-point scale on which 10 means that a task is impossible to perform without error, and one means that the task is easy to perform. Intermediate values on the scale correspond to increasing degrees of workload. In the scale's original applications, users' ratings were used as criteria for deciding whether a design or task needed modification. The higher the rating, the greater is the need to modify the design, or reduce the task demands.

A major virtue of the Cooper-Harper scale is that the wordings of the scale's items can be modified to make the scale explicitly applicable to the design feature or task whose workload demand is being assessed. Giving a concrete reference to the object of the rating scale reduces some the factors that would limit the reliability of the scale. If the wording modifications are done without violating the decision tree principles of the scale's structure, the modifications should not harm the validity of the scale.

---

[27] Based on the author's experience and opinion.

The factors that can affect the reliability and validity of the Cooper-Harper Scale are described later in Section 4.4.4.2, (page 197). These are factors that affect the reliability and validity of any subjective measurement scale, which the Cooper-Harper scale is, as indeed all workload measurement scales are.

A third and final type of workload measurement method is called the spare capacity method. This consists of measuring how much spare capacity a test participant has for performing a secondary task while performing the primary task whose workload demand is of interest. The spare capacity method presumes that a user has a fixed maximum work load capacity. If the user gives the primary task all the attention needed to perform it properly, then performance on the secondary task represents the workload capacity available for the secondary task. By subtraction, this gives the workload capacity taken up by the primary task. Thus, the more demanding the primary task is, the less capacity will the operator have for performing a secondary task.

For the following reasons, spare capacity is not a recommended method for measuring workload in a design test. The premise that primary and secondary tasks have additive workload demands is difficult to sustain. The fact that one can create secondary tasks that differentially interfere with the primary task is proof that the two workload demands are not necessarily additive [52]. In addition, design tests are usually aimed at representing realistic operating conditions. Having test participants doing secondary tasks that are unrelated to operations destroys the realism of the test conditions.

Overall, the best measures of workload are either measures derived from speed and accuracy measures, or measures based on workload scales like the Cooper-Harper scale. The spare capacity approach should be used only if there is no alternative.

4.4.4.1.3 Validate Candidate Measures with Test Results Users

Once the possible measures have been identified, the users of the test results should be consulted about what kinds of information they need from the test. Test results users include the PM, on the behalf of the system user organization, and the design team. Normally, if the test measures satisfy the CCIS HFE Requirements, they will also satisfy the PM. The design team, on the other hand, may have special information needs connected with how to revise the design if the test results turn up inadequacies in the design.

Ideally, a test measure's results will show both whether the design is adequate, and how to improve the design when the results suggest that improvement is needed. This is not always possible. Inadequate performance on a test measure may only show that a design feature is inadequate, but not how the design needs to be revised. Finding prescriptive test measures is always difficult - so difficult that insisting on having only prescriptive test measures would risk ending with no test measures at all. One way around a lack of prescriptive measures is to have more than one design alternative to test, or at least a set of predecessor system baseline measurements. With design alternatives, or a predecessor to fall back on, the design prescription is a matter of whatever design option yields the best results at the end of the test.

Whatever the means available to deal with prescriptiveness, it is important that the issue be dealt with early in the development of the test. This is best done by consulting with the design team to see how to construct the test to make the test results useful to the design team. Although ensuring that a design test produces prescriptive information is hardly the sole responsibility of the HFE specialist, the HFE specialist is responsible for finding out what information the design team needs to get from the test, and for designing the test to meet these needs. Equally important, the HFE specialist is also responsible for finding out what information the acquisition authority needs to get from the test.

Having identified candidate test measures based on the criteria described in this section, another set of criteria must be considered before making a final selection of measures for the test. These criteria concern the mathematical measurement properties of the test measures.

4.4.4.2 Selecting Measures According to Adequacy of Measurement Properties

The preceding section described test measure selection criteria defined by whether or not the measures would provide data pertinent to the purpose of the test: for example, whether or not the test measure would show the merits of a design to be tested. This section describes measurement selection criteria concerned with the measurement properties of candidate test measures: specifically, the validity and reliability of the test measures. By definition,

> *the validity of a test measure is the degree to which the measure measures what it is supposed to measure*, and
>
> *the reliability of a test measure is the consistency and precision with which the measure measures whatever it measures.*

As these definitions imply, having test measures that are both valid (i.e., measure what they are supposed to measure) and reliable (i.e., measure precisely) is a fundamental requirement that a design test must meet. If the test measures do not measure what they are supposed to measure, and are not accurate, they cannot yield any useful information about the merits of a design. Thus, validity and reliability represent criteria that any test measure must meet before it can be used in a design test.

4.4.4.2.1 Validity of Test Measures

Although validity means the degree to which a measure measures what it is supposed to measure, validity is also a relative quantity. Cronbach [102] pointed this out in discussing the validity of a standardized test,

> 'A test that helps in making one decision may have no value at all for another. This means that we cannot ask the question "Is this a valid test?" The question to ask is "How valid is this test for the decision I wish to make?" or "How valid is this test for the interpretation I propose for the test?"'

In a system design test, this quotation readily translates into the question, 'How valid is this *test measure* for the decision I wish to make, or the interpretation I propose for the test results?' Because validity is always about an interpretation or decision making context, any description of the validity of a specific test measure must include a description of the purpose for which the measure is being used. Another analogy with standardized tests makes a related point.

When a standardized test is created, its validity and its scoring norms are established by giving the test to samples of people known to have varying degrees of whatever the test is supposed to measure. Then, when a test user is considering using the test, he or she looks at the norms sample to see if the characteristics of the people in the norms samples are similar to the characteristics of the people that the test user plans to test. For example, if a test designed to predict academic achievement was validated on secondary school students, it will be valid for predicting the academic achievement of other secondary school students, but less valid for predicting the academic achievement of post secondary students. This is why the standards for developing standardized tests [103] require the test developer to put a full description of the characteristics of the norms samples in the test's user manual. This also implies that the persons responsible for conducting the design test have a duty to describe clearly the validity limitations of the design test when they report the design test results.

The example of validation norms samples implies that results from a design test are only valid as far as the design test reflects the system and operating environment about predictions are to be made. How well a design test measure reflects the eventual system environment depends on two things: 1) the design representation or testing medium used to administer the design test, and 2) the nature of the test measure. Normally, HFE experience can be used to estimate the impact of the testing medium on each measure. For example, user performance measures on a storyboard medium would have little validity or no validity at all. On the other hand, certain opinion measures might be just as valid on a storyboard as on a fully functional prototype.

Because validity is specific to context, there are several types of validity. In design testing, validity is essentially what is traditionally called content validity[28] (e.g., Cronbach [104]). For practical purposes, the content validity of a test measure is the degree to which the 'content' of the test measure is clearly related to the thing being measured. For example, if everyone connected with the project agrees that the shortness of the time taken to complete a planning cycle is a measure of command and control effectiveness, then the length of time it takes to complete a planning cycle is a *valid* measure of command and control effectiveness. Similarly, the time it takes test participants to complete a planning cycle when using the system's facilities is a valid measure of the system design's adequacy. The validity of every test measure may not be so clear-cut as in this example. Nevertheless, the validity of any test measure is how much a reasonable person would agree that the measure shows a design's adequacy, and agree that the measure is pertinent to the decision making purpose of the design test.

---

[28] The other major type of validity is called predictive (or sometimes called concurrent) validity. Predictive validity is a statistical correlation, for example, the correlation between students' scores on an academic achievement test and course grades in later schooling. Predictive validity seldom applies to design testing.

Parenthetically, the definition of content validity described here is somewhat cruder than the more elaborate definition given by Salvendy and Carayon [105], and others, who define content validity by reference to the collective content validity of a set of measures. However, design test measures are usually interpreted in terms of what individual measures indicate about individual design features. So, defining content validity in terms of individual measures is generally more relevant, and more intelligible to the users of the design test results.

4.4.4.2.2 Reliability of Test Measures

The reliability of a test measure can be expressed mathematically in several different ways. However, no matter how reliability is expressed, it depends on the error variance of the test measure; that is, variance owing to random factors that cannot be accounted for from other variables thought to influence values of the test measure. As this implies, reliability can only be measured empirically. The reliability of test measures can affect a design test in two ways. First, the more reliable a test measure is, the smaller the statistical confidence interval representing the precision of test measure is. Second, a more reliable test measure will need a smaller measurement sample to achieve a specified level of measurement accuracy. As this suggests, reliability is an important consideration in the experimental design for a design test. It will therefore be discussed in more detail in Section 4.4.5.6 (page 230).

Two sets of factors affect the validity and reliability of a test measure. One set contains factors associated with the instrument used to administer the test measure. These factors will be described in Section 4.4.5 (page 204). The other set contains factors representing the intrinsic nature of the test measure. These factors will be discussed next.

4.4.4.2.3 Validity and Reliability of Different Types of Test Measures

The intrinsic factors that affect validity and reliability depend on the nature of the test measure. Test measures used in design tests fall into one or the other of two basic categories: 1) objective measures of test participants' performance, and 2) subjective measures of test participants' opinions about the system design. Objective measures of performance fall into three categories: 1) speed of performance, 2) accuracy of performance, or 3) some combination of speed and accuracy. Measures of speed of performance include such things as the time taken to do a task or series of tasks, the rate of performance of a repetitive task, and the reaction time for responding to an event. Accuracy measures include such things as precision of performing a task, the number or size or the errors made in performing a task, and the correctness of judgments and decisions.

Because there is always a tradeoff between speed and accuracy in human performance, all objective performance measures are in some sense a combination of speed and accuracy. For example, if speed is used as a primary performance measure, steps have to be taken to ensure that accuracy is either held constant or taken into account. Conversely, accuracy is only meaningful when speed is held constant or taken into account.

As mentioned earlier, speed and accuracy measures are related to measuring the effects of workload capacity and workload demands. That is, workload capacity is by definition a function of the speed and accuracy of performance. Similarly, speed and accuracy are functions of workload demand. Thus, objective measures of performance in a design test can often be treated either as the amount of the workload capacity that the design permits, or as the amount of the workload demand that the design imposes on the user.

Subjective test measures are typically measures of users' opinions about various design features. The measurement validity of opinion measures comes from the fact that research has shown consistently high correlations between measures of user performance and user's opinions about a system [98]. Nevertheless, as will be shown later, subjective measures are not without validity weaknesses. User opinions can be gathered using paper and pencil questionnaires or face-to-face interviews. In some circumstances (circumstances to be avoided if possible), questionnaire items may have to be used as proxies for objective performance measures. For example, if getting direct measures of participants' speed of performance is impractical, getting estimates of speed of performance by asking participants questions in a questionnaire might be necessary.

Each of the two types of measures, objective and subjective, has it own validity and reliability characteristics. These characteristics are described next.

4.4.4.2.4 Validity and Reliability of Objective Test Measures

Objective test measures based on measures of speed and accuracy of user performance are generally intrinsically valid and reliable. If the measures identified for a design test are consistent with the requirements in the CCIS HFE Requirements, and are clearly related to needs of the design test, there should be few doubts about their validity. The reliability of objective performance measures involving measurement of speed and accuracy is typically more than adequate. Measures of speed are empirical quantities that are normally easy to measure accurately if proper measurement instruments are used. Measures of accuracy that are simple counts of errors are also quite reliable if the errors are conscientiously observed and recorded.

However, some accuracy measures, while being objective, are not necessarily as valid and reliable as other accuracy measures. An example is one type of measure specified in the CCIS HFE Requirements. This is, 'measures of users' awareness and comprehension of tactical situation and tactical events.' Awareness of the tactical situation could be measured using what would amount to a knowledge test containing items testing the participant's knowledge of the tactical situation. Using proper test construction methods like those described in Cronbach [104] and Lord and Novick [109], producing a knowledge test that is both valid and adequately reliable is not difficult. Failing to use proper test construction methods can easily lead to a knowledge test that is neither valid nor reliable.

Usually, objective measures are intrinsically valid if they are direct measures of quantities agreed to be valid measures of the test design's adequacy. Objective measures are also generally intrinsically reliable if they are properly constructed and administered in the design test.

4.4.4.2.5 Validity and Reliability of Subjective Test Measures

Subjective test measures are essentially opinion surveys. They can, for example, provide valuable information about participants' reactions to the design, along with participants' ideas about design improvements. The fourth set of questions in Table 12 (page 193) are examples of these types of questions.

Opinion measures are most useful when they are used as adjuncts to objective performance measures. They can provide a means of getting information about factors that affected participants' performance on the objective test measures. Open-ended questionnaires or interviews can be used to get participants' ideas about design improvements. In tests where the participants are members of the user organization, a platform for expressing opinions about the system design helps to give users a sense of participation in the system development process.

Although useful for design evaluation, subjective measures also have well-known potential validity and reliability weaknesses. These potential weaknesses are liable to become real when the opinion instruments are poorly or carelessly constructed. Fortunately, most problems caused by poor item construction can be prevented by using the questionnaire construction methods that will be described later (Section 4.4.5, page 204).

4.4.4.2.5.1 Validity of Subjective Measures

The potential validity weaknesses of subjective measures fall into two categories: 1) mismatches between what the questionnaire author intended to ask the questionnaire respondents[29] and what the respondents thought the author was asking, and 2) asking questions that respondents are not qualified to answer. The first validity weakness comes from a variety of questionnaire item[30] construction faults. These include using vague or ambiguous item wording, using language and technical terms that respondents do not understand, and, in closed ended questionnaire items, failing to provide item response alternatives that cover all the respondents' possible responses. Overall, poorly constructed questionnaire items can produce results that have little to do with respondents' actual views: in other words, results that are not valid representations of the respondent's views.

The second potential validity weakness of subjective measures (asking questions that respondents cannot answer accurately or reliably) is not so easy to prevent. Questionnaire respondents generally do their best to answer whatever the items in the questionnaire ask. This results in

---

[29] For consistency with the literature, the term respondent will be used to refer to a person answering questions on a questionnaire. However, in the context of design testing, a respondent is also a test participant.

[30] Like the term respondent, the term item is the common technical term for a question on a questionnaire. The term item is used because questionnaire items are not always literally questions.

respondents attempting to provide answers to questions that they are not really qualified to answer. In a design test, test participants are not normally qualified to answer direct questions about the merits of design features. Test participants are not design experts. In contrast, test participants are qualified to give opinions about whether or not they found a design feature convenient and easy to use during the testing activities. Even here though, the validity of the respondents' opinions might be questioned if they had no experience with systems similar to the test system. Overall, participants are qualified to answer questions only about things that are within the area of their subject matter knowledge. Questions about matters outside the participants' expertise are not valid questions and do not yield valid data.

Even on questions that test participants might be thought to be qualified to answer, validity may still be weak because of response biases in the participants' responses. Going back to the example of participants' opinions about whether or not a design feature was convenient and easy to use, interpreting participants' responses will still be difficult if there is no independent way of knowing what participants understand by the term, "convenient and easy to use." In addition, system users' opinions about such things as their ability to operate a system are often inconsistent with their objectively measured ability to operate the system. Comparisons between opinion and measured performance show that users usually overestimate their abilities [106]. Overall, self-reports of behaviour, abilities and attitudes typically include biases that are hard to correct for without independent means of measuring the sizes of the biases. Uncorrected biases reduce the validity of any measure that relies on self-reports.

4.4.4.2.5.2 Reliability of Subjective Measures

The factors that limit the validity of subjective measures, that is, inadequate item wording and response bias, can also limit reliability. Inadequate item wording was just mentioned as a cause of reduced validity. This is so when all respondents misinterpret an item's wording in the same direction. So, respondents answer the item consistently, but answer something different from what the questionnaire author intended to ask. This is equivalent to a constant error of measurement.

When poor item wording causes different respondents to interpret an item's meaning in different ways, the result is lowered reliability owing to an increased error variance. When respondents get different understandings of what an item is asking, the variance of the responses will be larger than it would be otherwise. This variance will appear in the data analysis as error variance.

Response bias was also mentioned as a factor in the validity of self-reports of system operating abilities. Since response bias represents a form of measurement error, it is also represents a reduction in reliability. Response bias is present to some degree in all questionnaire data. It comes from the common human desire to be seen favourably. In a design test this bias usually emerges as a general tendency to give more positive opinions about system features than participants' would give outside the confines of the design test. In the design test environment, participants' good will towards the system developers often leads to giving favourable opinions

about the system design. Participants may also feel, sometimes justifiably, that they are under pressure to give favourable opinions about the system.

4.4.4.2.5.3 Avoiding the Limitations of Subjective Measures

Although proper questionnaire construction methods can minimize the effects of the factors that limit validity and reliability of questionnaire data, good questionnaire construction can seldom eliminate these factors. The only way to truly minimize the effects of factors that reduce the validity and reliability of opinion data is to avoid relying on opinions as the sole means of evaluating a system design. Instead, subjective measures should only be used with objective measures, and then, only to amplify or clarify the results from subjective measures.

Although a design test that relied solely on questionnaire data would not meet the intent of the CCIS HFE Requirements, there may be considerable temptation or pressure to rely exclusively on questionnaires. Questionnaires seem easy and cheap to construct and administer, and they appear to non specialists to provide understandable measures of what test participants think of the system design. The attractiveness of subjective measures is also subtly reinforced by test participants' response biases favouring the system design. Because of these pressures, HFE specialists need to take extra effort to caution the users of test results about the limitations of subjective measures.

Objective measures can be used to minimize the validity and reliability limitations of subjective measures in three ways.

1 Do not use a subjective measure where an objective measure would be practical to construct and administer.

2 Give the user an objective frame of reference for responding to each subjective measure: for example, by:

   2.1 giving participants enough knowledge of results to allow them to give informed responses to subjective measures, or

   2.2 explicitly referencing the subjective measure to specific system users' tasks.

3 Ensure that all subjective measures have one or more objective measures that can serve as validity criteria in the data analysis. If, for example, participants' opinions can be shown to be consistent with objective performance results, then the opinions are likely valid.

In general, using subjective measures to amplify and clarify the results of objective measures is the proper use of subjective measures, and the best way to avoid their validity and reliability shortcomings.

4.4.4.3 Final Selection of Test Measures

The final selection of measures should at this point be measures that meet three criteria:

1 meet the requirements specified in the CCIS HFE Requirements;

2 meet the specific information needs of the design test, and

3 promise to provide adequate validity and reliability for the purposes of the design test.

Perforce, individual test measures selected according to these criteria will provide information about individual design features, or information about specific user tasks, rather than information about the system design as whole. Although providing information about individual design features is usually the priority of a design test, considering the selected test measures as an aggregate is usually also worthwhile. This consideration may suggest what the aggregate would have to say about the design as a whole. It may also show that some of the measures are redundant and could be eliminated. Conversely, it could show that gaps are leaving possibly important design features untested. It might also show incomplete groupings of measurements that could provide useful information if they were filled out by the addition of one or two other measures. Finally, considering the selected measures as a whole will show whether or not the overall measurement procedure is practical.

Once the collection of test measures has been reviewed, and modified if necessary, construction of the test instruments can begin. The construction of measurement instruments for a design test is described in the next section.

4.4.5 Constructing Test Measurement Instruments

Test measurement instruments are the media or devices used to collect measurement data during a design test. Once a test measurement has been selected, the next task is to construct an instrument to collect the data. Constructing a measurement instrument includes three tasks: 1) identifying or devising the measuring instrument, 2) defining how the instrument will be administered during the test, and 3) defining how data from the instrument will be recorded. An additional consideration in instrument construction is at what points and how often the instrument will be administered during the design test.

The goal of instrument construction is to maximize the instrument's validity and reliability while minimizing the cost of the instrument. The following subsections describe instrument construction procedures designed to achieve this goal. The procedures are described separately for test measures based on direct observation and test measures based on questionnaires. This division corresponds roughly to the objective-subjective division used in the previous section to describe procedures for specifying test measures. The only difference will be that questionnaire construction will include construction of objective instruments such as paper-and-pencil tests of knowledge and questionnaires about factual information to be provided by test participants.

Before the instrument construction procedures are described, the next section briefly discusses how instrument measurement reliability affects test costs. The cost of running a design test is a direct function of the reliability of the test measurement instruments. Thus, instrument reliability is an important consideration in all aspects of instrument construction.

4.4.5.1 Instrument Reliability and Design Test Costs

In a measurement application, the number of samples needed to achieve a given degree of measurement precision depends on the measuring instrument's error variance. The greater the instrument's error variance, the greater will be the number of measurements needed to achieve a specified precision of measurement. Precision being reliability, and reliability being a function of error variance, this means that to achieve a specified level of precision, an unreliable instrument requires a larger sample size than a reliable instrument does.

This relationship between reliability and sample size means that instrument reliability has a substantial impact on the cost of a system design test. A design test done with unreliable measuring instruments will require larger sample sizes to achieve the same measurement precision as could be achieved with reliable instruments and smaller sample sizes. Larger sample sizes translate into requiring more test participants, and having more and longer test sessions than would be needed otherwise. More participants and more testing will mean higher cost. Thus, instrument unreliability represents a substantial direct cost in design testing. Conversely, having reliable instruments can represent a considerable cost saving.

The relationship between cost and instrument reliability will be described in more detail in Section 4.4.5.6 (page 230). For the present, the relationship means that investing the effort needed to construct reliable test measurement instruments has a real and favourable cost impact that is worth keeping in mind throughout the instrument construction procedures described in the following sections. In particular, investing effort in selecting and constructing good instruments pays off in reduced costs for running the system test.

4.4.5.2 Construction of Direct Observation Instruments

Direct observation instruments include instruments such as:

1 an observer using a stop watch to record the times that test participants take to perform various tasks;

2 an observer using a checklist to record the occurrence of errors made by participants while performing tasks;

3 software built into a system prototype to record participants' speed and accuracy data automatically;

4 subject matter experts evaluating the accuracy or quality of participants' performance, and

5 post test observations from video or audio recordings of participants' performance.

The following subsections describe instrument construction for the different types of instruments represented in these examples.

4.4.5.2.1 Data Collection by Observers

Presuming that the time and error measures will be valid by definition, the only instrument construction concern will be to ensure their reliability of the collected data. The reliability of measures based on an observer's recordings of times and errors depends on three factors:

1 the observer's training for taking and recording the observations;

2 ensuring that the observer will have adequate time and opportunity to make the observations during the test, and

3 provision of appropriate forms for recording data, and adequate procedures for collecting completed data recording forms and entering the data into a computer for data analysis.

Data collection by observers can be treated as a type of questionnaire data. Looked at in this way, the data collection form is essentially a questionnaire that an observer fills out at specified points in a design test exercise. Thus, a good deal of the procedure in Section 4.4.5.3, page 210 (Construction of Questionnaire Instruments) can be applied to constructing direct observation instruments and direct observation administration procedures.

If observers are given appropriate data collection forms, proper training, adequate opportunities to make observations, and dependable data handling procedures, the reliability of the observer-recorded data should be more than adequate for the needs of the test.

4.4.5.2.2 Automated Data Recording

Software built into a system prototype to record participants' speed and accuracy data automatically is often an attractive possibility [107]. Automated data collection will produce data at least as reliable as data collected by observers. However, before adopting automated data recording, its advantages and disadvantages should be weighed against those of alternative manual data collection methods. The advantages and disadvantages of manual and automatic data recording are listed in Table 14.

Although automated data collection has several advantages, it is not necessarily easy to set up. It may require significant programming effort, and will not be easy to modify if there is a change of mind about what data to collect. Possibly the biggest disadvantage of automated data collection is the fact that the automated

Table 14 Advantages and disadvantages of manual and automated data collection methods.

Manual Advantages:
- cheap and easy to design and implement,
- flexible and easy to modify,
- easy to trace and correct recording errors,
- data collection can be monitored as it happens,
- high reliability.

Manual Disadvantages:
- labour intensive data collection,
- labour intensive data processing,
- restricted to low data rates and volumes,

Automation Advantages
- labour saving data collection,
- labour saving data processing,
- high data rates and data volumes,
- high precision.

Automation Disadvantages
- expensive and difficult to implement,
- low flexibility and difficult to modify,
- difficult to trace errors,
- difficult to monitor, and often unreliable,
- limited to recording system events.

data collection system will still depend on manual data collection methods. This is because the automatic data collection system can only record system events, but not necessarily what the test participant was doing when each system event was recorded. This means having a parallel manual data collection system to record test conditions and user behaviour. In addition, the data analysis will require linking the automated data to the manual data.

As a labour-saving device, automated data collection is usually only justified when it can be used over an extended period, or over a series of design tests where the extended usage generates enough labour savings to recover the initial cost of the automation. In one-time tests, the data collection labour savings will seldom justify the cost of automation. This suggests planning on using manual data collection. Automated data collection can be reserved for situations where data cannot be collected manually, or where building automated data collection into the prototype to be used for the design test would be easy to do.

Although programming automated data collection into a test system is not usually practical, other means of automation are often possible. Data can sometimes be tapped from the computer running the test design prototype without requiring modification of the prototype software. The prototype's platform may automatically log data that would be useful. The designers may be recording data for use in design - data that might be also useful for the HFE design test. Portable computers can often be used in data collection: for example, for observers to record data that would otherwise be recorded on paper forms. As will be mentioned in the next section (Section 4.4.5.3), administering questionnaires online or on a portable computer is sometimes convenient.

4.4.5.2.3 Collecting Subject Matter Experts' Evaluations

Measures of command and control effectiveness might include evaluations of test participants' performance in tasks such as producing a tactical plan, recognizing or responding to tactical events, and so on. Subject matter experts' evaluations can be vulnerable to the same validity and reliability to which all subjective measures are vulnerable. This is because a subject matter expert's evaluation or opinion often contains a subjective element.

Applying the following principles will help to achieve valid and reliable evaluations from subject matter experts.

1 The subject matter experts should be clearly qualified to act as experts, and be free from potential biases such as having involvement is the system development project, or having unique prejudices about command and control philosophy.

2 At least two, preferably more, subject matter experts do each evaluation.

3 The subject matter experts must be given a common set of evaluation dimensions. The dimensions should be based on objectively observable participant behaviours or work products. Similarly, the valuation dimensions should include objective criteria for gauging adequacy of participant performance. To the extent possible, the evaluation dimensions and criteria should be quantifiable, and suitable for post test calculations of reliability.

4 The subject matter experts must be trained to do the evaluations. The training should include practice with realistic scenarios in which subject matter experts can compare their evaluations, and by doing so, converge on a common set of criteria for evaluating participants' performance.

5 The subject matter experts should be provided with forms convenient for recording their evaluations, and convenient for later computer data entry.

6 During test sessions, the subject matter experts must do their evaluations *independently*.

Abiding by these principles will ensure that the subject matter experts' evaluations achieve maximum validity and reliability. Having more than one expert doing evaluations independently will make it possible to calculate the reliabilities of their evaluations. This will give a basis for assessing how much confidence to put in the evaluation results.

These principles also show that achieving useful evaluations from subject matter experts takes a substantial effort. However, nothing is gained by doing things inadequately: say, by bringing in an individual subject matter expert to make evaluations without a systematic framework for doing the evaluations, and without proper indoctrination into the purpose and nature of the evaluations.

4.4.5.2.4 Data Collection with Video Or Audio Recordings

Video or audio recordings are a possible way collecting performance data. Participants' behaviour can be analysed later by recording test measurement data from viewing the tape. However, as discussed earlier about using video recordings in task analysis (see Section 4.2.5.4.1.3, page 87) video tapes must be viewed to be analysed. This may take three to 10 times longer than it took to record the tape. Unless there is a multiple-camera setup, the video recordings will not show everything that is going on in the test. Then, with multiple cameras, there will be more tape to view.

Generally video and audio records are inefficient. Good test measures do not require continuous recording of participants' behaviour. They only require taking samples of behaviour at scheduled times or on the occurrences of particular events. A human observer making *in situ* observations according to a predefined sampling plan will normally be far more efficient than having an equivalent observer spooling through many hours of video tape to pick off the specified observations. Because of the generally lower fidelity that recordings will have compared with first-hand observation, the reliability (and possibly validity) of data from recordings is bound to be lower than that from direct observation. Therefore, recordings offer no measurement advantages over direct observation.

One potential use of recordings is to record critical incidents that occur during the test, and to provide clarification of anomalies in the observed data. Recordings can also serve as a backup record in case observed data are lost or observations are missed. Recordings can also be used in debriefing participants: for example, for getting a participant's explanation of an unexpected event during a test session. Finally, recordings may be useful for post test briefing of design personnel

and acquisition authority personnel. All these uses of recordings do not cost much, and generally do not interfere with the running the test.

4.4.5.2.5 Summary of Construction of Direct Observation Instruments

The following summarizes the construction of direct observation instruments.

1 Collection of objective performance data by observers is a valid and reliable means of collecting data if observers:

- 1.1 are trained for recording the observations;
- 1.2 have adequate time and opportunity to make the observations, and
- 1.3 have appropriate forms for recording data, and
- 1.4 adequate procedures for subsequent data processing have been prepared.

2 Automated data recording can provide data at least as reliable as data collected by human observers. However, automated recording has disadvantages, the most serious of which are the cost of developing it, and the difficulty of linking with data recorded by other means such as observers.

3 Subject matter experts can provide valid and reliable measures of command and control effectiveness if

- 3.1 the subject matter experts are clearly qualified, and are free from potential biases;
- 3.2 two or more subject matter experts do each evaluation;
- 3.3 the subject matter experts have a common set of evaluation dimensions based on objectively observable behaviours or work products, and have similarly objective criteria for gauging adequacy of participant performance;
- 3.4 the subject matter experts are trained to do the evaluations and get systematic practice doing the evaluations in realistic scenarios;
- 3.5 the subject matter experts should be provided with appropriate data recording instruments, and
- 3.6 the subject matter experts do their evaluations independently.

Video and audio recordings are normally inefficient and not particularly reliable ways of collecting observational data compared with direct observation methods. However recordings can be useful for such things as clarifying anomalies in observed data, recording critical incidents, and debriefing test participants.

Overall, observers and subject matter experts provide valid and reliable test measurement instruments if they are supported with proper training and data recording devices. Automated data collection is valid and reliable when its cost disadvantages can be overcome. Video and audio recordings have limited usefulness as performance measurement instruments.

4.4.5.3 Construction of Questionnaire Instruments

A questionnaire instrument is anything that involves asking a respondent to answer a set of questions about some topic. A questionnaire instrument can be a paper and pencil questionnaire, a face-to-face interview questionnaire, a computer-administered questionnaire, or any other device that provides a means of asking questions and recording answers. Only two conditions make an interview necessary: 1) the respondent has to be shown materials that cannot be put in a paper questionnaire, or 2) the questionnaire contains open-ended questions, or questions that an interviewer would need to explain to the respondent. Questionnaires used for design tests rarely have either of these requirements.

The construction of a questionnaire is largely independent of the medium that will be used to administer the questionnaire. The following will describe the construction of a self-administered paper-and-pencil questionnaire because this is by far the most common medium for use in a design test. Construction variations needed for other media are usually self-evident.

In a design test, questionnaires can be used for any of the following purposes:

1 to get information about test participants' demographic status, technical and military qualifications, experience, and any other background information that might be pertinent to understanding the results from the test;

2 to get test participants' opinions about the design being tested, or about any matters that might influence the design;

3 to assess test participants' on things such as: knowledge of how to operate the system, and comprehension of the tactical situation during a test scenario[31], and

4 to get participants' estimates of subjective experiences such as the amounts of workload experienced under different test conditions.

Although the questionnaire instruments that can be used in a design test vary in their purposes and contents, the procedure for constructing a questionnaire is much the same no matter what the type of questionnaire. Table 15 contains the steps that Sudman and Bradburn [108] recommend for constructing a questionnaire. The following subsections describe the different steps of the construction procedure.

[31] A questionnaire assessing a respondent's knowledge or some other capability is conventionally called a test rather than a questionnaire. However, for convenience, the term questionnaire will be used here for all instruments that, in effect, ask test participants to answer questions.

Table 15 Steps for developing a questionnaire.

1	Make a list of the information to be gathered.
2	Look for examples of other questionnaires containing examples of items or item formats that could be used in the questionnaire.
3	Draft the questionnaire items and organize the items into logical groups and sequences.
4	Format the questionnaire.
5	If the results are going to be analysed on a computer, set up column and coding definitions for keypunching.
6	Have the draft reviewed by co-workers, or any other individuals who are qualified to comment.
7	Revise the draft and have it reviewed again.
8	Prepare procedures and instructions for administering the questionnaire.
9	Pilot-test the questionnaire on a small sample of respondents who are similar to the respondents who will eventually be answering the questionnaire.
10	Obtain comments from pilot test respondents and questionnaire administrators.
11	Revise the questionnaire and administration procedures.

4.4.5.3.1 Drafting the Questionnaire

Drafting the questionnaire covers the first five steps in Table 15. The first of these steps is to list the information to be collected in the questionnaire. The test measures selected for the design test should provide the main source for making up the list. If the questionnaire is a knowledge test, subject matter experts will likely have to be asked for information: for example, for a list of skills and knowledge required for doing a task.

A common problem in constructing a questionnaire is failing to think of all the information topics that the questionnaire should cover. If the list of information is incomplete at the start of the questionnaire construction, awkward questionnaire reformatting will be needed to insert new items when the missing information is finally noticed; worse, the missing information may not be noticed until the data are analysed and it is too late to do anything about it. To avoid this problem, it is a good idea to use a source of topics that provides a systematic and comprehensive scheme for identifying topics. For example, if the questionnaire concerns users' operating tasks, referring to the task descriptions of all the user tasks supported by the design would ensure that the questionnaire covered all tasks.

Once the list of information to be covered in the questionnaire is complete, work can start on constructing the questionnaire's items. This work can begin by converting the information requirements into questions, and then translating the questions into appropriate item formats. Item construction effort can often be reduced by using items from existing questionnaires, or by using existing questionnaires for item ideas. Besides saving labour, ready-made items and item formats, by the fact that they have been used before, come with some assurance that they will

work as intended in the questionnaire that is being constructed. Previously published questionnaires may provide entire questionnaires, or scales like the workload scales described in Section 4.4.4.1.2 (page 194). Sources for existing questionnaires include colleagues, research reports, and general literature.

An example of a ready-made set of questionnaire materials is a set developed by DCIEM. This is a system of questionnaire and focus group investigation methods designed to identify operational item deficiencies, elicit user feedback, insights, and requirements for items in the acquisition cycle, and to track the operational effectiveness of acquired items once in service. The system, ACCESS (for Army Combat Clothing and Equipment Survey System), offers a layered approach to the acquisition of user feedback. Each layer, consisting of a preliminary survey, a detailed questionnaire, and focus group investigations, focuses the direction and efforts of each subsequent layer. Commercially available computer scoring methods are used to improve speed, accuracy and efficiency during data acquisition, reduction, and analysis[32]. Information on ACCESS is available from the Operational Human Engineering Group at DCIEM.

Questionnaire material that appears in the open literature is not usually copyrighted, unless a copyright is made explicit. Explicitly copyrighted questionnaire material must not be used without permission.

When items are constructed from scratch, one or more of the standard item formats should be used. A variety of established item formats are available in the literature. Table 16 lists the most common formats, along with examples of their uses. Each item format in Table 16 has two parts. One part is called the stem. The stem contains the question being asked. The second part of the item is the set of response alternatives that the respondent can choose from to answer the question in the stem.

One or more of the formats in Table 16 should adequately cover all the questions likely to be asked in a design test questionnaire. Using an established format is always preferable. An established format will generally guarantee that the item will work as intended. Chances are that respondents will already be familiar with the format, and in fact, may expect to see it used. A familiar format makes it easier for the respondent to understand what the item is asking, and easier to give an answer. New or unusual formats will require more pretesting than established formats.

[32] David Beevis personal communication. Information on ACCESS is available from the Operational Human Engineering Group at DCIEM.

Table 16 Types of questionnaire item formats and examples of their uses.

Item Format	Examples of Uses
Multiple choice allowing selection of one response alternative	Knowledge tests, forced choice preference selection questions, demographic questions
Multiple choice allowing selection of more than one response alternative	Questions about acceptable alternatives, demographic questions
Rating scale questions asking for a response on a fixed interval numeric scale	Opinion questions asking for a quantitative level of preference or agreement
Point estimation questions, for example, number of hours of operating experience	Information questions asking for estimates of frequency, duration or probability of an event
Rank ordering items in which the respondent ranks a set of items according to preference	Questions about preferences or utilities (in paired comparison and multidimensional scaling applications)

The following subsections describe how to construct items according to the formats in Table 16.

4.4.5.3.1.1 Constructing Multiple-choice Items

Multiple-choice items that ask the respondent to supply information about himself or herself are generally easy to construct. About the only thing that requires extra care in composing this type of item is to make sure that the response alternatives cover all the possible responses that the respondent could give. For example, the response alternatives for an item asking a respondent's highest level of education must include all levels of education. With very few exceptions, multiple-choice alternatives should include an 'Other' or 'Don't know' alternative in case an alternative was missed.

Multiple-choice knowledge test items require much more careful construction than other types of multiple-choice items do. A multiple-choice knowledge test item has a stem comprising the knowledge question being asked, and (usually) four response alternatives, one of which is the correct response, and the remainder of which are incorrect responses. The incorrect responses are called distractors. The following rules should be followed in creating a knowledge test item.

1 The stem of the item should ask about a straightforward fact or concept that the respondent could be expected to know. The wording should be as brief as possible, preferably one sentence, and not more than two.

2 Knowledge test items should have four response alternatives. Four response alternatives usually provide a reasonable level of item difficulty, and reasonable protection from respondents guessing the correct answer.

3 The distractors must be at least plausible correct answers. More than one distractor can to be a correct answer if the response that will be marked as 'correct' is the best or most correct answer.

4 All the response alternatives should be as short and concise as possible, and of the same length. When composing response alternatives, there is a natural tendency to make the correct response alternative longer than the distractors. Respondents will intuitively recognize that the longest response alternative is probably the correct answer.

5 Getting the correct response on the item must not depend on having got the correct response on another item.

6 The correct response must be verifiably correct either by reference to some form of documentation or by reference to subject matter experts.

When a knowledge test is used, the data of interest are usually respondents' total scores on the set of items making up the test. Thus, the validity and reliability of the test scores will depend on the composite or collective properties of the items making up the test. Two factors affect validity: 1) the validity of each item on the test, and 2) the comprehensiveness of the test. To ensure validity, each item must sample some aspect of the knowledge domain covered by the test, and all the items taken together must cover all parts of the knowledge domain being measured.

The reliability of a test is a primarily a function of the length of the test. To be reliable, a knowledge test usually needs to have at least 50 well-constructed items. Although tests that are shorter than this can have good reliability, it would be unwise to count on this unless prior information on the reliabilities of the individual items can be used to estimate the test's reliability ahead of time. This prior information will not normally be available for a system design test. Therefore, 50 items should be treated as a minimum requirement.

The validity and reliability of composite scores is a complex subject. Lord and Novick [109], and Cronbach [104] provide in-depth treatments of the subject.

4.4.5.3.1.2 Constructing Rating Scale Items

A rating scale item asks the respondent to use a numerical scale to represent the strength of his or her opinion about some subject. Figure 25 shows an example of a rating scale item. The stem instructs the respondent to use the numeric scale to express how easy the respondent found it to retrieve a message. The instruction in parentheses tells the respondent how to make a response. When the questionnaire contains a series of similar rating scale items, this instruction can be given at the beginning of the series instead of being included in every item. The scale at the bottom of the figure shows the numeric values of the scale, and labels to indicate the dimension represented by the scale, and the direction of magnitudes on the scale.

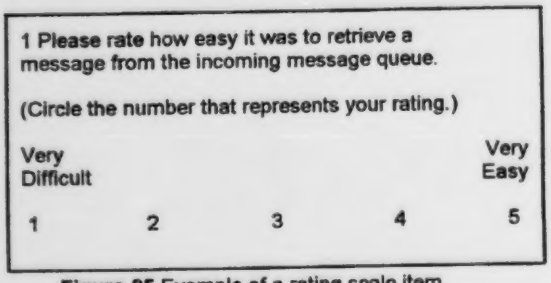

Figure 25 Example of a rating scale item.

Rating scales can be used to get expressions of opinion on a variety of dimensions: ease of use of a system function as in Figure 25, strength of agreement with the statement in the item stem,

perceived likelihood of making an error when using a system function, or any dimension of opinion that can be represented as a quantitative variable. Rating scale items can also be set up as a matrix convenient for getting ratings of several different objects simultaneously: for example, all the tasks associated with a specific operational sequence or a segment of a test scenario. Figure 26 shows an example of such a matrix.

Please rate how easy it was to do the following.	Very Difficult				Very Easy
1 Retrieve a message from the incoming message queue.	1	2	3	4	5
2 Create a reply to a message.	1	2	3	4	5
3 ...					

Figure 26 Example of a matrix for rating multiple items.

Most people are familiar with rating scale items and can respond to them with very little instruction. The following are rules that should be followed in constructing rating scale items.

1 For bipolar dimensions (such as Very Easy *versus* Very Difficult) use a scale with an odd number of values so that the middle value represents a neutral opinion. For unipolar dimensions (such as 'How would you rate the readability of the User Manual?') use a scale that runs from zero to some maximum, for example, zero to 10.

2 Define the high values on the scale as positive or strong opinions, and low values as negative or weak opinions.

3 Choose a scale with a number of values that will represent all the likely gradations of respondents' opinions. A five-point scale is usually a safe choice. Use a higher or lower number of points if more or fewer gradations of opinion are likely to be needed. If the size of scale to use cannot be decided from prior knowledge, pretesting is the only way to confirm if the size of the scale is correct.

4 If possible, use the same numeric scale throughout the questionnaire, or at least for all items within the same section of the questionnaire.

5 Like the example in Figure 25, give the respondent definitions for the meanings of the highest and lowest values on the scale.

6 The object or concept to be rated should be simple and concrete. The wording of the rating item should be similarly simple and concrete. Ratings of complex objects or abstract ideas are likely to be highly variable, reflecting different respondents' different understandings of what is to be rated.

7 Every item should be checked to ensure that the wording in the item stem is consistent with the labels on the rating scale. Inconsistency between the wording of the item stem and the labels on the scale is a common and easily made error in constructing rating items, so a special check for discrepancies is always worthwhile.

If the rating scale data are going to be analysed with statistical methods that include the assumption that the data come from continuous linear variables, it is important that the rating scale items support this assumption. Research shows that this assumption is valid for many types of human judgments. However, research also shows that the underlying mental scale can be different for different types of judgments, and that the scale is not always linear. Using careful item construction methods is the first step to ensuring that the rating data will sustain the mathematical assumptions in the later statistical analyses.

4.4.5.3.1.3 Constructing Point Estimation Items

A point estimation item asks the respondent to provide an estimate of the magnitude of a physical variable that the respondent is expected to know or can estimate. The variables include such things as time, distance, frequency of an event's occurrence, and the probability of an event. A point estimation item can be constructed either as a multiple-choice item, or as an item that asks the respondent to supply a specific value. A multiple-choice format for asking the respondent's age would give response alternatives consisting of age ranges such as 21 to 30, 31 to 40, and so on. An item asking for a specific value would simply ask the respondent to supply his or her age.

Point estimate data have several limitations. The subjective scale that a person uses to estimate a quantity depends a great deal on the type of quantity being estimated. Thus the accuracy with which people can provide estimates of quantities depends a great deal on the quantity being estimated. For example, people generally know their age very accurately, but may know how often they do a particular work task only approximately. The linearity of subjective scales also depends on the quantity being estimated. The subjective scale for estimating distance is generally arithmetically linear, while estimates of the frequency of occurrence of events are generally quite nonlinear. Subjective scales may also have constant errors, including response bias errors caused by the universal desire to be seen favourably.

Doing the following in constructing point estimation items will help to overcome and take account of the limitations of subjective estimates of quantities.

1 Specify a scale with logical physical units that the respondent can use for making the estimate. For example, when asking for the frequency of occurrence of an event, specify once an hour, once a day and so on. Do not use an indeterminate scale like very often, often or seldom.

2 Use a scale with a unit that is clear and familiar to the respondent. For example, specify nautical miles if the nautical mile would be the natural unit for the respondents to use.

3 Since many subjective scales are not linear, make the intervals in response alternatives correspond to equal intervals on the subjective scale. If the shape of the subjective scale is not known from prior information, pretest the item to get an idea of the shape of the scale.

4 Try to anticipate response biases by wording the item to counteract the bias.

Following these rules will yield data that are suitable for any statistical analyses that depend on having data that meet the assumptions for continuous variables. Even if the underlying data do

not represent linear scales, the data can still be transformed to meet the assumptions of the statistics, provided that the form of the scale is at least approximately known.

4.4.5.3.1.4 Constructing Rank Order Items

Item ranking items are used to get a respondent's order of preference for a set of items. Rank ordering items are of two sorts: 1) simple ranking of a set of items, and 2) paired comparison ranking. For the first form, the respondent is given a list of items to rank, and asked to supply ranking values: say, one to eight if there are eight items to rank. In the paired comparison form, the respondent is presented with every possible pairing of the items, and asked to tell which member of each pair is the most preferred.

Rank order items can be used to get preference estimates for objects whose qualities vary along several different dimensions. If the objects to be ranked are properly chosen, ranking data can be analysed to identify what dimensions the respondents were using to rank the objects. The analysis can also estimate the relative weights that the dimensions were given in the ranking process. This analysis is called multidimensional or multi-attribute analysis. It can be used to study behaviour associated with such things as perceived utility, decision making, and risk taking [111].

The construction of ranking items is generally straightforward. For simple ranking, the construction of the item consists of an instruction on what dimensions or attributes are to be used in the ranking, a list of the items to be ranked, and spaces for writing in rank values. Respondents should not be asked to rank more than about eight items. When have more than eight items to rank, they will find it hard to keep track of the items, and they will make mistakes such as using the same rank more than once. This mistake can be avoided by putting the items on cards that the respondent can sort according to his or her rank preferences. The items to be ranked should be mutually exclusive, and be things likely to represent a wide range of relative preference.

For paired comparison ranking, an item consists of three parts: 1) an instruction about what dimensions or attributes to use for the ranking, 2) a list of all possible pairings of the items to be ranked, and 3) spaces for writing in the preferred member of each item pair. Since the number of possible pairings of items increases geometrically with the number of items, paired comparisons can be used only when the number of items is small. Although paired comparisons are time consuming when there are more than a few items, they are generally easy for the respondent to do, and therefore more reliable than simple ranking.

4.4.5.3.1.5 Constructing Items in Other Formats

The general rules for constructing items in a nonstandard format are the same as those described for constructing items in the standard formats:

1 The item should deal with a simple fact or concept and include:

1.1 a concise description of what is being asked,

1.2 instruction on how to respond to the item,

1.3 a convenient way to give the response, and

1.4 formatting to minimize respondents' marking errors and clerical errors in subsequent data processing.

.2 The item should be based on an established psychological or psychophysical construct such as the fact that people can generally quantitatively express opinions and preferences.

3 The construction and content of the item will support the assumptions underlying the planned statistical analyses of results.

Any item construction format that meets these formats is suitable for a questionnaire.

4.4.5.3.1.6 Open-ended Items

An open-ended item is a question that asks for a verbatim written or spoken response. Open-ended items should be avoided because they are difficult to collate and analyse. For analysis, responses to open-ended questions have to be transcribed and then classified into groups representing common sets of responses. This is a time-consuming job that rarely yields information that could not have been gathered by closed questions.

If respondents are expected to have things to say that are not covered by the questionnaire items, space can be put aside in the questionnaire for respondents' comments. If comments are welcome, directions for adding comments should be put in the questionnaire instructions.

4.4.5.3.1.7 Minimizing Response Bias Effects

Response bias is a tendency for a respondent to bias answers in a direction that puts the respondent in a favourable light. Excepting knowledge test items and basic demographic items, nearly all questionnaire items are subject to some response bias. In a system design test, the two strongest sources of response bias are typically the desire 'to look good', and a real or imagined desire to give positive opinions about the system being tested.

Although eliminating response bias is probably impossible, several things can be done to reduce it. These are listed in Table 17.

The first item in Table 17 is part of a general need to give participants assurance that all

Table 17 Ways to minimize response bias.

1 Give a written commitment that the responses will be kept confidential, and that they will be used only for the purposes of the research.
2 For interviews, have the interview administered by someone who will be seen as independent and a peer of the respondent.
3 Make the questionnaire items refer to specific objects, times, or situations.
4 When dealing with sensitive or threatening topics, include qualifiers in the item stem to recognize and moderate the respondent's sensitivity.
5 Make sure the response alternatives cover the full range of possible responses.
6 Give enough information to ensure that the respondent fully understands the context of the question.

individual results collected in the test will be kept confidential. Provisions for confidentiality are described more fully in Section 4.4.5.12 (page 246).

The ways to reduce response bias by means of item wording are limited only by the questionnaire author's ingenuity. For example, transferring the subject of the item from the respondent to a group as in, 'Do you think that the training most members of the squadron have received will allow them to do this task effectively?' is one such device. An entertaining compilation of all the standard devices for reducing response bias by item wording is given in an often quoted article by Barton [110]. Barton illustrates the devices with an example question that simultaneously shows that getting around the problem of response bias is also sometimes impossible. A survey question similar to Barton's original example would be, 'Have you robbed a bank?' Among Barton's examples, an instance of using a qualifier (see paragraph 4 in Table 17) to reduce response bias would make this question, 'Many people rob banks. Have you ever robbed a bank?' Barton's example is described in full in Sudman and Bradburn [108].

As discussed earlier in Section 4.4.4.2.5.3 (page 203), the most effective way to avoid or minimize response bias effects is to use objective measures wherever possible, and to avoid using subjective measures as the sole means of drawing a conclusion about a system design proposal.

4.4.5.3.1.8 Formatting the Questionnaire

Once the items have been drafted, the questionnaire can be formatted. Because the effectiveness of a questionnaire depends not just on the contents of the items, but also on how the questionnaire looks to the respondent, the draft questionnaire should be formatted to look like a final version.

The first step in formatting is to organize the items into logical groups and sequences. The best organization will correspond to whatever organization would make the most sense to the respondent. Proper organization will help the respondent to focus attention and remember the information needed to answer each question, and generally minimize the effort needed to complete the questionnaire.

It is well worth the effort to make the questionnaire look like a professionally produced document. A good-looking questionnaire shows respect for the respondent, and a serious interest in what the respondent has to say. Proper formatting will reduce errors in marking answers to items. It will also reduce errors when the completed questionnaires are processed for data analysis. Since questionnaires are often administered at the end of a design test exercise section when test participants are looking forward to a rest, minimizing the effort needed to complete the questionnaire is important to getting the test participants' attention. Table 18 lists basic guidelines for formatting a questionnaire.

Assuming that the questionnaire results will be analysed on a computer, formatting the questionnaire includes setting up numeric coding schemes for the responses to the items. For multiple-choice items, this is normally done by numbering each response alternative on each item. The computer data for a respondent's answer to a multiple-choice item is then the number of the

response alternative that the respondent chose for answering the item. For rating scales, the computer data are the rating numbers chosen by the respondents. For any item that asks for a numeric value, age for example, the computer data are whatever numbers the respondent supplied when he or she answered the item.

Formatting the questionnaire includes making provision for keypunching data from the completed questionnaires. For the most part, a format that makes it easy for the respondent to answer the questions will also make it easy to read the responses and enter them on a computer. Generally, the most effective way to support data entry work is to arrange the questionnaire items so that the responses are easy to pick out from the item stems and respondent instructions.

Table 18 Guidelines for formatting questionnaires.

1 Use large type, and allow plenty of space for the items.

2 Use a different type face, or some other device such as parentheses, to identify directions to the respondent or the interviewer.

3 Number the items. Use letters to identify supplementary items, or sub-items.

4 Set up the questions and response alternatives in a regular vertical or columnar format.

5 Code the responses for computer analysis, and define the columns and fields that will be used to set up any computer files. Plan how the questionnaire responses will be keyed.

6 Use coloured paper to identify different sections of questionnaires, or different versions of the same questionnaire.

7 Use rugged binding to prevent pages coming loose during the multiple handlings that a questionnaire goes through. A booklet is easiest for the respondent or interviewer.

The computer file for storing the response data does not have to be set up until a completed questionnaire are ready for data processing. However, setting up the file when a first draft of the questionnaire is ready will help to ensure that the file will be correctly set for the questionnaire data. The documentation for whatever statistical data analysis package is going to be used for data analysis will provide directions for the file setup. The file for storing questionnaire data is normally a text file. The coded questionnaire responses are usually stored as a packed set of numbers with no separator codes. The data for each respondent can be stored as one or more lines (or records) in the file. Data for a particular questionnaire item are identified by the line and column numbers in which they are stored in each respondent's file record. All statistical data analysis packages can read this file format, if the package is given the line and column locations of the data for the different questionnaire items. Some packages have data entry facilities that can be tailored to the questionnaire.

4.4.5.3.2 Questionnaire Review and Pilot Testing

The following describes the last six steps in constructing a questionnaire, steps 6 to 11 in Table 15 (page 211). These steps are the following.

1 Have the draft questionnaire reviewed by co-workers, or any other individuals who are qualified to comment.

2 Revise the draft and have it reviewed again.

3 Prepare procedures and instructions for administering the questionnaire.

4 Pilot-test the questionnaire on a small sample of respondents who are similar to the respondents who will eventually be answering the questionnaire.

5 Obtain comments from pilot test respondents and questionnaire administrators.

6 Revise the questionnaire and administration procedures.

The next subsections describe these steps.

4.4.5.3.2.1 Reviewing and Revising the Draft Questionnaire

Having a draft reviewed by colleagues will provide good feedback for improving the wordings of items, improving the questionnaire format and organization, and generally suggesting changes that would make the questionnaire more convenient for the respondent. After revising the questionnaire, the reviewers should review it again. Though one review-revise-review cycle should be enough, the cycle should be repeated until the reviewers are satisfied.

At this point the draft questionnaire should be pilot tested on a small sample of respondents who are representative of the people who will be the test participants. The comments from these respondents will provide the basis for the final version of the questionnaire. For a design test questionnaire, the pilot test sample may have to answer some items about design test exercises that the pilot sample will not have experienced. If so, it will not be possible to pilot test these items until the pilot test of the complete design test. On the other hand, if the questionnaire pilot test respondents can at least give comments on the wordings of these items, the items should be kept in the questionnaire pilot test.

If the questionnaire review process includes a review by the users of the test results, this review should be done only after the questionnaire pilot test and the final draft of the questionnaire is complete. This avoids getting the results-users involved in the technicalities of the questionnaire's construction, and allows them to focus on whether or not the questionnaire covers everything they want covered.

4.4.5.3.2.2 Design the Questionnaire Administration Procedure

Planning for administering the questionnaire includes:

1 scheduling the time when the questionnaire will be administered during the test sessions;

2 arranging for a space where the respondents will complete the questionnaire, and

3 specifying the procedure for administering the questionnaire, and collecting the completed questionnaires.

The time required to complete the questionnaire should be not more than twenty to thirty minutes. Beyond thirty minutes, the quality of respondents' answers is bound to deteriorate. If the questionnaire takes longer than thirty minutes, it should be broken up so that it can be completed in separate sessions, with another activity, or a rest between sessions.

The time for completing the questionnaire should be scheduled as part of the design testing sessions. Test participants cannot be expected to spend their time off doing questionnaires. Questionnaires are usually given at the end of test sessions. Therefore, the test session schedule must allow adequate time for the questionnaires and not force participants to miss rest breaks or work overtime. If test participants perceive that completing the questionnaires is not an activity as important as other test activities, they will not give a full effort to answering the questionnaire.

Respondents should be given a quiet space such as a conference room or classroom for completing the questionnaires. The questionnaire administrator should be present to answer any questions that the respondents have about filling out the questionnaire. The administrator may also need to ensure that respondents answer the questions independently, and do not discuss the questions among themselves until they have completed their questionnaires.

When the respondents have completed the questionnaires, the administrator should collect the questionnaires, and if possible, check that they have all been completed. Ensuring that participants put their names or identifiers on completed questionnaires is particularly important. The completed questionnaire should then be stored in a safe place.

The procedure for administering questionnaires should be a written instruction containing:

1 the time or occasion when the questionnaire is to be handed out;

2 in detail, the instructions that the questionnaire administrator is to give to the respondents on how to complete the questionnaire;

3 the location where the questionnaire is to be administered;

4 how much time the respondents are to be given for completing the questionnaire, and

5 detailed instructions on collecting the completed questionnaires, checking the questionnaires for completeness, and storing the questionnaires.

For a questionnaire administered by an interviewer, there is an additional test administration procedure: training the interviewer.

Training consists of briefing the interviewer on the purpose and nature of the questionnaire, and giving the interviewer practice at administering the interview questionnaire. By the end of training the interviewer must meet the requirements listed in Table 19. An interview should flow like a conversation. This may require some extra work on the wordings of the questions so that they sound natural when spoken by the interviewer.

A common problem in interview questionnaires is making sure the interviewer deals with

Table 19 Requirements for conducting an interview.

1 Read the questions in a clear conversational manner.
2 Read the questions exactly as they are worded.
3 Maintain a neutral unassuming manner throughout the interview.
4 Avoid prompting the respondent or doing anything that might bias the respondent's answers.
5 Correctly and neatly fill in the respondent's responses.

response alternatives correctly. Some types of items are designed so that the interviewer will read only the question and mark which response alternative the respondent gave. In other items, the interviewer reads out both the question and the response alternatives. Interviewers can easily get confused and read out response alternatives when they are not supposed to be read out, or fail to read out the response alternatives when they are supposed to be read out. Putting everything the interviewer is supposed to say in a distinctive typeface will help the interviewer to avoid this error.

Completion of the questionnaire administration procedure completes the construction of the questionnaire.

4.4.5.3.3 Summary of Questionnaire Construction

The following summarizes the steps in the construction of a questionnaire.

1 List the information to be collected.

2 Draft the questionnaire using item formats appropriate for the types of information to be gathered.

3 Format the questionnaire so it is professional looking and easy to use.

4 Have the draft reviewed and pilot tested as extensively as possible before administering a final version.

5 Design the questionnaire administration procedure including provisions for scheduling, facilities for administering the questionnaire, and collection and storage of the completed questionnaires.

Careful questionnaire construction is critical to getting valid and reliable results.

4.4.5.4 Construction of Expert Panel Decision Instruments

A typical expert panel is a group of subject matter experts employed to give expert opinions about the relative merits of a set of decision alternatives. Meister's [62] formal function allocation procedure using weighted ratings of allocation criteria to evaluate design alternatives is an example of the use of such an expert panel. Expert panels can be used to choose among options in many situations: for example, to decide which of several off-the-shelf purchase options best meets requirements, or to decide which of several system development options to pursue, or to evaluate the tactical plans produced by several teams in a command and control test exercise. Generally, a decision choice panel may be useful whenever the decision involves complex decision alternatives and many factors that must be considered in making the decision. In fact, this represents the decision making environment of any high level command and control organization.

Expert panel applications have five features in common. The features include

1 several experts, typically fewer than 10, but occasionally many more,

2 several decision alternatives that must be ranked according to an overall order of merit,

3 several decision criteria that are usually a mix of objective and subjective variables on which each decision alternative can be rated,

4 for each criterion, a weight representing the importance of the criterion in the decision, and

5 an algorithm or formula for combining the experts' ratings into a single value representing the overall merit of each decision alternative.

In Meister's function allocation procedure, the formula giving the merit of a decision alternative is the sum of the products of the criterion weights and the corresponding average of the expert ratings on the decision criteria. The simple weighted mean for getting aggregate measures on decision variables is only one of many possible approaches to defining composite decision choices (see, for example, von Winterfeldt and Edwards [111], and Lehto [64]).

Expert panel rating procedures based on some form of multidimensional scaling metric can be powerful tools for evaluating decision alternatives. Nevertheless, they are also vulnerable to misuse. What makes them vulnerable to misuse is that their apparent mathematical rigour can conceal a variety of serious measurement inadequacies. These potential measurement weaknesses can be seen by considering the underlying reliability and validity of the variable representing the decision criterion, that is, the reliability and validity composite variable that represents the overall merit of each decision alternative.

The factors that affect the reliability and validity of a decision criterion depend on the particular multidimensional scaling metric used to define the decision criterion. The following factors are ones that are common to most multidimensional scaling metrics, and ones that need to be dealt with when applying an expert panel decision choice procedure.

1 *Reliability factors.*

1.1 The panel's sample of subject matter experts must represent the population of subject matter experts that could be drawn on to form a panel.

1.2 The panel members must be unbiased in their partiality to any of the decision alternatives.

1.3 The panel members must all have the same breadth of expertise, or each member's judgments must be weighted according to the member's area of expertise.

1.4 The weights for the criterion variables must be defined independently of the ratings given by the subject matter experts. This rule would be violated, for example, if the same panel members defined the weights and did the ratings. If the weights and ratings are dependent, then the weights and ratings will likely be correlated, thus violating the linearity assumptions that most decision metrics require.

1.5 Panel members can do their ratings either independently, or together to produce consensus ratings. Although independent ratings are the norm and avoid the complications of interactions among panel members in a consensus process, either approach is valid. However, whichever approach is taken, it must be applied to the

panel members consistently, and it must be consistent with the multidimensional scaling metric used to define the composite decision criterion.

1.6 A questionnaire is the natural vehicle for eliciting panel members' ratings. This questionnaire requires the same questionnaire construction effort described in the previous section (Section 4.4.5.3, page 210).

1.7 After the panel does its ratings, the consistency of the experts' ratings (or criterion weights) must be assessed and shown to be consistent across experts. If the experts gave significantly different patterns of ratings, there is good reason to go back and revise the rating procedure, and certainly good reason to question the reliability of the panel's results.

1.8 The rating questionnaire or procedure should contain mechanisms to permit showing whether or not each subject matter expert gave consistent weight or rating values when the objects of the weights or ratings were similar. If each expert was not consistent in his or her responses, there is again good reason to question the reliability of the results.

1.9 Unless the behaviour of the multidimensional scaling metric is very well known, its sensitivity should be tested by seeing what happens to the results when small changes are made in the values of criterion weights, or when one or a few rating variables are removed from the composite decision criterion. If the composite criterion is unduly sensitive to these small changes, the reliability of the results is again questionable.

2 *Validity factors.*

2.1 The primary basis for the validity of a subject matter expert panel's results lies in the degree to which the panel members are truly experts in the decision problem. Subject matter experts for a panel must be chosen with the same care as they are chosen for task analysis work (see Section 4.2.5.4.1.5, page 91).

2.2 Just as lack of homogeneity in the panel members' expertise can limit reliability, it can also limit validity. Thus, for the panel's results to be valid, the experts must be homogeneous in their expertise, or measures must be taken to account for differences among the subject matter experts' fields of expertise.

2.3 When available, another test of the validity of an expert panel's results is whether or not the results are consistent with the results of similar studies. Results might be inconsistent with other studies for many reasons, some or all of which might be spurious. Nevertheless, if the results are inconsistent, the reasons need to be looked into.

2.4 A final test of validity is whether or not the results fit with common sense, or with a global estimate of what the best decision alternative would be. For this, the value of the expert panel decision process is often not so much in what decision alternative it comes up with, but in showing why the chosen alternative is the best one.

There are plenty of precedents for using systematic quantitative procedures for exploring multiple-choice decision problems, but constructing a valid and reliable procedure does require careful and considerable preparation effort. Therefore, applying a quantitative procedure to a decision problem should be done only after a careful consideration of what will be involved in

doing it properly. Done improperly, a quantitative decision making procedure may amount to little more than a contrived exercise in comparing apples and oranges.

4.4.5.5 Design Test Medium

A design test requires some form of design representation to serve as a medium for getting test participants' reactions to the design. Thus, in a design test, test participants either 1) inspect a representation of the test design and give opinions about the likelihood that the design will satisfy requirements, or 2) use a design representation to do simulated operations tasks. In the first case, participants' opinions about the design are a measure of the design's utility. In the second case, the design's utility is measured by how well test participants can perform operations on the design representation.

The remainder of this section is in two parts. The first describes the different types of test media available for doing design tests. The second discusses how the choice of a design test medium affects the validity and reliability of test measurements.

4.4.5.5.1 Types of Design Test Media

Table 20 lists the different types of design representations that can be used as media for a design test. The first column of the table lists the media roughly in order of fidelity with a final system design. The remaining two columns describe each medium and how it can be used in a design test.

In CCIS development, drawings might be used to get opinions about a design concept from the user organization's subject matter experts. Although user subject matter experts can provide useful opinions about the general system engineering merits of a concept design, it is unlikely that they could provide any significant opinion about HFE matters that an HFE specialist could not provide independently. Therefore, drawings are not likely to be used in any design test requiring significant HFE participation.

The storyboard can be useful for getting users' opinions about such things as the flow of work in a proposed system, and the layouts of control consoles and physical facilities. Storyboards were first used in advertising to illustrate the appearance of proposed television commercials. They are also widely used in movie making to plan filming. As these applications suggest, the storyboard is an effective medium for representing anything that can be conveyed as a picture-illustrated story. In a command and control context, a storyboard can be used to represent a mission profile. In fact, graphical representations of mission profiles often amount to storyboard representations. Although a storyboard is a comparatively cheap medium for illustrating a design proposal, its effectiveness is only as good as its fidelity with the design being represented. To be effective, a storyboard requires good illustration drawings or photographs, and a carefully designed layout.

Table 20 System representations and their applications in HFE design evaluations.

Test Media	Description	Application
Drawings	Schematics, layout diagrams or flow charts showing a preliminary design or system concept.	Evaluation of a design concept using subjective measures such as questionnaires.
Storyboard	A series of drawings and illustrations showing how system functions would work.	Conducting walk-through demonstrations and using checklists or questionnaires for design evaluation.
Mockup	A fairly realistic but non-functioning physical model of the system's physical facilities.	Evaluation of physical facilities using walk-through demonstrations, questionnaires, and possibly limited performance measures for evaluation.
Prototype	A function model of all or part of the system capable of supporting user operations of interest to the design test.	Detailed or advanced design evaluation with selected operations scenarios and using both user performance measures and subjective measures.
Advanced Prototype	A fully functional accurate model of the final system design capable of supporting all user operations.	Evaluation of the final design with realistic operations scenarios and using both user performance measures and subjective measures.

mockup is useful for testing the design of a physical facility such as a headquarters or a control centre. It is particularly useful for facilities where personnel interact with one another, and personnel frequently move about the facility. Beevis *et al.* [66] contains a good example of the use of a mockup of a ship bridge for a design evaluation. A mockup of a physical facility can be produced cheaply with light plywood and paste-on drawings to represent control and display surfaces. If the physical dimensions accurately represent the real facility, a mockup can provide a quite realistic rendition of how the proposed design would look.

Instead of building a physical mockup, there are computer-based media that can be used to create a form of mockup. A CAD (Computer-aided Design) program can produce a variety of three dimension views of a facility. CAD representations are inexpensive to produce, especially if a CAD program is being used for design anyway. Animated 3D software can be used to provide realistic tours of a proposed facility. Finally, a mockup can be produced as a virtual reality rendition that would permit a visually realistic experience of a facility. However, animated 3D and virtual reality representations are expensive to produce compared with the design testing value that could be recovered. Moreover, animated 3D and virtual reality would not be as flexible as a traditional mockup, and except portability and small space requirements, would offer no advantages over the traditional mockup.

For testing the computer-based components of a CCIS design, the software prototype is the obvious choice for testing a design. Prototypes are generally cheap and easy to produce, and are in fact produced almost as a matter of course in software development. Thus a software prototype can be produced at almost any stage of design, including the idea design stage. A reasonably functional software prototype can be used for design reviews and for design testing.

Although, software prototypes are comparatively easy to produce in theory, they are not always easy to exploit in practice. Designers are often, naturally enough, reluctant to release a prototype until the design is perfected. This cannot only represent a delay in getting access to a prototype for testing, but also result in design progressing too far to allow design changes when a test is finally conducted. Lack of computer resources is another factor that can limit access to a prototype. For example, if the design system is a network, and the prototype is only functional when operating as a network, getting access to the prototype for many HFE activities will likely be difficult.

If the prototype requires substantial computer resources, or can function only in some form of large-system mode, consideration should be given to creating a purpose-built standalone prototype for HFE design review and design testing. Much of the HFE work, including design testing, concerns single-user functions that can be simulated on a modest prototype.

Another common practical difficulty with using prototypes in design testing concerns how well the prototype's performance matches the performance of the planned system. Some prototypes are essentially notional representations that are adequate for giving a good idea of what the planned system will be like. A notional prototype might, for example, give a fairly accurate representation of the planned system, but be much slower than the planned system, and have graphics displays with much lower resolution than the planned system's graphics. In a demonstration, most people can extrapolate from the prototype's appearance and performance to imagine the final system's appearance and performance.

Extrapolation would not be possible in a design test based on measures of test participant performance. In a design test, participant performance results from a prototype can be generalized, but they cannot be extrapolated. In particular, if participant performance results from an inadequate prototype fall below design requirements, performance results would not necessarily be any better on the 'improved' final design system. In addition, lack of a degree of functionality in a prototype can have a variety of complex interactions with participant performance variables. For example, a slow system response time may not just slow the user, it may also change the whole character of the user's performance.

A final practical consideration in using a prototype in a design test is to ensure that everyone understands that the purpose of the test is to predict potential users' performance on the final system; it is not to test the system hardware and software. This may be a subtle distinction, but it can often have a significant impact on the running of the test. Differences in perceptions of the purpose of the test often emerge as unreliability in the prototype's operation. If, for example, the prototype's software crashes because it cannot handle the test participants' usage, the software engineers may consider this to be useful information. In contrast, test administration personnel and participants might see the crash as anything from a minor to major nuisance depending on how long it takes to recover from the crash. If the prototype software has not been (software) tested on users before the design test, a variety of software failures during the design test will be quite likely. As useful as these failures might be for software engineering purposes, they can seriously damage or even destroy design test.

In a design test, it is important that the prototype function reliably. If it does not, testing activities may be seriously disrupted. Moreover, the malfunctions will inevitably have a spurious bad influence participants' opinions about the system design. The only way to verify that a prototype will function reliably during the design test is to do a pilot test before running the design test proper.

4.4.5.5.2 Test Medium Impact on Measurement Validity and Reliability

Although the fidelity of the test medium clearly has an impact on the validity and reliability of the test measures used in a design test, quantifying this impact is seldom possible. With validity, clearly, any test medium that is less than a fully functional prototype will cause test measures to be less valid than they would be with a higher fidelity test medium. Recalling the earlier discussion of Cronbach's definition of validity (Section 4.4.4.2.1, page 197), the validity of a test medium should be consistent with the importance of the decisions that will be made from the test results. If important decisions are being made from the test, then a high fidelity test medium is required. Recalling the corollary of Cronbach's definition, the report of the test results must clearly describe the degree to which the nature of the test medium limits the validity of the test results.

The effect of test medium fidelity on test measurement reliability is usually not so serious a concern as fidelity effects are on validity. This is largely because adverse reliability impacts can be overcome in the experimental design of the design test. If a measure has inadequate reliability, the inadequacy can be cured by taking a larger sample of measurements. Possibly the best way to analyse the potential impact of test medium fidelity on measurement reliability is as follows.

The effect of test medium fidelity on reliability can be understood from the idea of a test measurement's error variance. Reliability is an inverse function of a test measures error variance. Error variance is a sum of the error variances of the measurement's sources of error. The fidelity of the test medium is one of these sources of error variance. Therefore, the impact of fidelity on a test measure's reliability depends whether the nature of the test medium combined with the nature of the test measurement would be a significant source of error variance. For some measures fidelity, or lack of it, will probably have no effect. For example, on an opinion measure, if the test medium provided a system representation that was equally accurate for all the test participants, the error variance owing to fidelity would be virtually zero.

An example where fidelity would very likely contribute to error variance is a performance measure based on how long it took a test participant to find an object in a map display. For this example, suppose that the prototype display is a low resolution display, while the planned system display is going to be a high resolution display. Here, performance on the prototype will have higher error variance than it would on the final system because of the lower display resolution on the prototype. Since the lower display resolution is tantamount to the prototype's lack of fidelity with the target system, the error variance can be interpreted as an effect of test medium fidelity. In this particular case, the lack of fidelity will not cause not just higher error variance. It will also cause a constant error owing to slower overall performance on a low resolution display than on a high resolution display.

The effect of test medium fidelity on reliability is usually difficult to quantify. A rare exception is where there happen to be empirical data available for how the test medium fidelity would affect reliability. However, an informed judgment can usually be made about how test medium fidelity will affect specific test measurements. Then, the test sample size can be increased to nullify the effect, or the test medium can be modified to improve fidelity.

In summary, the effects of test medium fidelity on test measurement validity and reliability can be dealt with by applying the following three principles.

1 The impact of fidelity on measurement validity should be treated as a matter of the importance of the decisions that will be made from the design test results. If important decisions are to be made, the test medium fidelity should be as high as possible.

2 The effects of test medium on reliability can be dealt with by using experience to judge the likely impact of fidelity on the different test measures. Then if necessary, the measurement sample can be increased to compensate, or the test medium can be changed to improve fidelity.

3 The users of the design test results must be told about any limitations that test medium fidelity will, or did, impose on interpretations of the results of the design test.

4.4.5.6 Design Test Experimental Design

The statistical design of a design test specifies four sets of quantities:

1 the number of test participants;

2 the numbers and sequences of testing conditions that make up the test;

3 the numbers of times and intervals at which test measures are to be administered, and

4 any other sampling quantities that would affect the statistical analyses of the test results.

The aim in creating an experimental design for a design test is to find an optimum balance between the cost of the design test and the confidence that the decision makers can place in the conclusions drawn from the results.

The idea of decision making confidence can be illustrated with the error in the mean of a sample of repeated applications of a test measure. The confidence that one can have in the accuracy the mean from a sample of measurements depends on the measurement error associated with the sample mean. The error of a mean depends on two variables: 1) the test measure's inherent error of measurement, and 2) the size of the sample used to get the mean of the test measure. The effect of the inherent error of measurement is that the larger the inherent error of measurement is, the larger the error of the mean is. Sample size has the opposite effect. The larger the sample size is, the smaller the error of the mean is. As this implies, no matter how large a test measurement's inherent error of measurement is, this error can be compensated for by averaging over a large enough sample of repeated measurements. This in turn implies that a test result can be evaluated to whatever level of confidence one wants by making the sample size large enough.

Since the cost of taking measurements increases as the sample size increases, there is a tradeoff between test cost and the confidence that can be placed in the test results. Thus creating a statistical design for a test consists of determining the measurement sample sizes that will optimize this tradeoff. The remainder of this section describes procedures for using available information about the errors of measurement of the test measures to determine an optimum sample design. Sample size determination is a complex subject that is too broad to cover in detail here. Therefore, the following will be limited to describing the principles underlying sample size determination, and to providing references to relevant literature.

The procedures will be described separately for two types of measures: 1) simple single-variable measures such as time and accuracy of performance measures, and 2) composite measures consisting of such things as the score on a knowledge test or the average of a set of rating scale items. Following the descriptions of sample design procedures, the section ends with a discussion of how to reconcile sample design requirements with project resource constraints.

4.4.5.6.1 Determining Sample Size for A Single-variable Measure

The confidence that can be placed in a test result is normally expressed as a statistical confidence interval. As a reminder, the confidence interval for the mean of a sample is given by:

$$-1.96 \leq \frac{\overline{X} - \mu}{S_X} \geq 1.96 \qquad 1$$

where $\bar{x}$ is the sample mean, μ is the population mean, and $s_{\bar{x}}$ is the standard deviation of the sample mean. The standard deviation of the mean is often called the standard error of the mean. The standard deviation of the sample mean is given by:

$$S_{\bar{x}} = \frac{S_x}{\sqrt{N}} \qquad 2$$

where s_x is the sample standard deviation, and N is the sample size.

The first equation says that chances are 95 in 100 that μ falls in the range $\bar{x} \pm 1.96\ s_{\bar{x}}$. The second equation says that $s_{\bar{x}}$ is an inverse function of the square root of N. The expression $\bar{x} \pm 1.96\ s_{\bar{x}}$ is the confidence interval, usually called the 95 percent confidence interval. Taken together, the two equations say that the larger the sample size is, the smaller the confidence interval is.

Figure 27 shows an example of how the confidence interval varies with sample size. The graph in Figure 27 shows the upper and lower bounds of the 95 percent confidence interval for a sample in which the mean (of a composite score) is 50 and in which the standard deviation is 20. Thus, for a sample of 10, the confidence interval is about 50 ± 12. If the sample size were 50, the confidence interval would be about 50 ± 6.

The graph in Figure 27 is a slight approximation because it implicitly assumes that the distribution of the sample mean is normal. This is not strictly true for all sample sizes. Conventionally, the t distribution is used to calculate confidence intervals for samples smaller than 30. If the t distribution had been used to calculate the boundaries for the smaller sample sizes, the confidence intervals would be slightly wider than shown in the graph. Despite the approximation, the graph still illustrates two general effects of sample size: 1) the size of the confidence interval decreases as the size of the sample increases, and 2) there are diminishing confidence interval improvements from increasing the sample size. These two general effects of sample size apply to nearly all statistical quantities that one might collect from a design test or any other behavioural experiment.

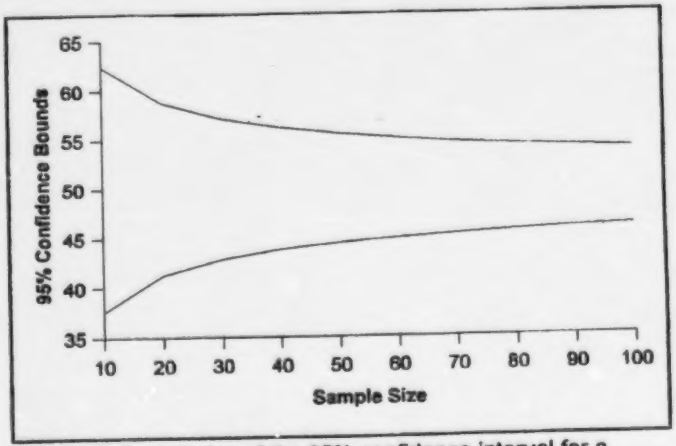

Figure 27 Boundaries of the 95% confidence interval for a mean of 50 and a standard deviation of 20.

Since the cost of sampling data is usually an increasing function of sample size, an ideal sample is one in which the size of the sample is *just* large enough to produce a confidence interval that is small enough for the needs of the experiment. For example, in Figure 27, if the experimenter wants to be 95 percent certain that μ is between 44 and 56, a sample size of 50 would be just right. A smaller sample size would not be large enough to provide 95 percent confidence. A larger sample would mean spending money on collecting unneeded data.

Calculating the optimum sample size for a statistic like the mean, of course, begs the question of what would be the optimum confidence interval. Determining the optimum value for a statistical decision criterion such as a confidence interval is a matter of considering three factors (Sudman [112]):

1 the decision maker's existing degree of uncertainty;

2 the gains and losses associated with the decision alternatives, and

3 the effect of uncertainty on the decision.

In statistics, uncertainty is the variance or the standard deviation of the statistic. Thus, in the example in Figure 27, the standard deviation used in calculating the confidence interval is essentially a measure of prior uncertainty about the variability of the mean. With no prior information, one must assume that the standard deviation will be equal to the population standard deviation. With prior information, one can assume that the standard deviation will be smaller than the population standard deviation. In some situations, a Bayesian approach can be used to calculate a standard deviation that takes account of prior information. In other situations, the

standard deviation has to be 'guesstimated' from combining informed judgment and available statistical data.

The gains and losses associated with the decision alternatives determine whether it is worth doing a study to get more information before making a decision. If the gains and losses are small, the cost of the study may be higher than either of the potential gains or losses. If so, the study would not be worth doing. If the potential gains and losses are large compared with the cost of a study, the study becomes a worthwhile investment.

The effect of uncertainty on a decision is determined by where the gains and losses' breakeven point stands compared with the prior expectation of the value of the statistic that will be used to make the decision. In the example in Figure 27, suppose that the breakeven point for a decision is such that if the population mean is 44 or greater, the gains from the decision will equal the losses. Now, an observed mean of 50 from a sample size of 50 or so would let the decision maker be 95 percent certain that the outcome of the decision would be profitable. This is a case where collecting a sample of data to estimate the population mean would be worthwhile, if the cost of collecting the data would not wipe out the profits. The study would not be worthwhile if the prior expectation of the value of the population mean was much greater than 44, or much lower than 44. That is, for example, if the decision maker already has a good idea that the population mean is well above 44, it is extremely unlikely that the study will show that the population mean is really below 44. Overall, the amount of new decision making information that a study can contribute is a maximum when the breakeven point is equal to the prior expected value the decision criterion.

The three factors that determine the economics of a study sample (decision maker uncertainty, decision outcome gains and losses, and study outcome uncertainty) can be expressed as a set of variables. Taking as an example a decision that will depend on the value of the mean found in the proposed study, the variables are:

1 the prior expected value of the study sample mean,

2 the gains and losses from the decision outcomes, and the gains and losses breakeven point, and

3 the prior expected value of the variance of the study sample mean.

Values of these variables can be combined with study cost information to work out an optimum sample size. This is a sample size that yields the best balance between the cost of the study and the value of the decision making information to be gained from the study. Sudman [112], applying a method developed by Schlaifer [113], gives an example using these variables to calculate an optimum sample for a survey. Sudman provides additional examples for determining sample sizes for a variety of other sample design problems. Kish [114] also contains examples of sample optimization problems.

Although most of the literature on sample optimization deals with survey applications, the statistical principles underlying sample design are the same no matter whether the sample is for a survey, a laboratory experiment, or a system design test. Therefore, if the statistical assumptions

remain valid, any technique developed for survey samples will usually apply just as well to design test problems.

Simulation is another way to explore sample design problems. Simulations can be done by two methods. One is by applying analytic methods, from Sudman for example, using different assumed values for the statistical and decision utility variable. The other simulation method consists of producing Monte Carlo simulations of raw data. For example, the following function is useful for generating normally distributed data.

$$X = \mu + \sigma\sqrt{-2\ln(R)\cos(2\pi R)} \qquad 3$$

X is the raw data variable whose values are to be generated. μ and σ are the mean and standard deviation of X. The quantity R is a random number between zero and 1.0. The quantity ln(R) is the natural logarithm of R. Equation 3 can be used in a spreadsheet to generate normally distributed data, which can then be analysed in various ways to explore the possible outcomes of the design study. Monte Carlo simulations can be used not only to explore sample size problems, but also to explore design questions such as arrangements and sequences of test conditions, and so on.

Whether one uses an analytic or a Monte Carlo method to determine a sample size for a measurement, the accuracy of the sample size determination will depend on the amount of prior information available for making the determination. Sometimes, data from prior studies or from the literature can provide almost all the information needed. In other cases, there will be little prior information, so defining parameters for the sample size determination will involve a good deal of educated guesswork. Although having to use guesswork is less satisfactory than having prior information, a systematic consideration of the optimization problem is always better than simply picking a sample size out of the air.

The preceding dealt with single-variable measures where the measure's inherent error of measurement is a fixed quantity. The next section deals with composite measures where the error of measurement is not a fixed quantity, but instead, is a quantity that can be controlled by the way the composite is constructed. Consequently, the construction of the composite is something that can be taken into account in determining the sample size. Conversely, impact on sample size requirements can be taken into account in constructing the composite measure

4.4.5.6.2 Determining Sample Size for A Single-variable Measure

Equation 4 gives the standard error of any measure that consists of composite[33] measure such as the score on a knowledge test, or the average score on a set of questionnaire rating items.

---

[33] For present purposes, a composite measure is the sum of scores on a set of items that all measure the same thing.

$$s_e = s_x\sqrt{1 - r_{xx}} \qquad 4$$

where s_e is the standard error of the composite score x, s_x is the standard deviation of the scores, and r_{xx} is the reliability of the composite score x. The reliability of a composite measure is a correlation coefficient that has a minimum value of zero and a maximum value of 1.0. Thus, from Equation 4, the higher the reliability of the composite score is, the smaller its standard error is.

The reliability of a composite score is an index of the degree to which all the items making up the composite measure the same thing. One standard way of measuring reliability illustrates this property. This is the split-half method, and it is the correlation between scores on half the items in the composite and scores on items in the other half of the composite. Other ways of measuring the reliability of a composite score involve using internal correlations among the items making up the composite [104] [105] [109].

Equation 4 illustrates the importance of reliability in determining the error of measurement of a composite score. Since the reliability of a composite score depends on the inter-correlations among items, the reliability of a composite can be maximized by constructing the composite from items with high inter-correlations. Methods for selecting items to maximize reliability are called item analysis methods. They are described in literature on psychological test construction (e.g., [109]).

Another way to increase the reliability of a composite is to increase the number of items in the composite. Figure 28 illustrates the change in reliability for two unit-length composites as the length of each composite goes from half its original length to half again as long as its original length. The upper graph is for a composite whose unit length reliability is 0.80. The lower graph is for a composite whose unit length reliability is 0.50. Thus, for example, if a composite made up of 50 items had a reliability of 0.80, decreasing the number of items to 25 would decrease its reliability to about 0.66. Increasing the number of items to 75 would increase its reliability to about 0.86. As the shapes of the graphs show, the incremental increase in reliability diminishes as the length of composite is increased.

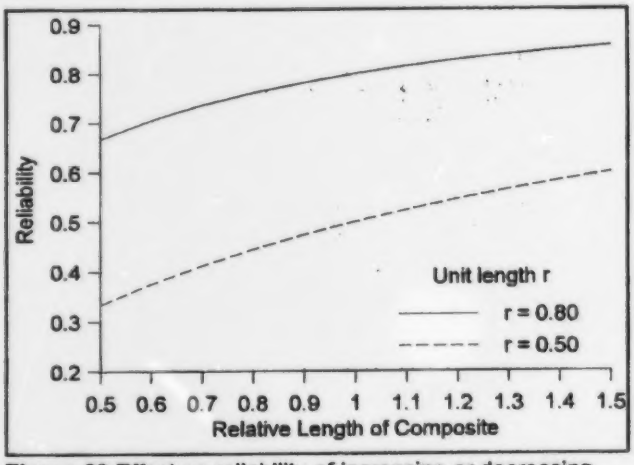

Figure 28 Effect on reliability of increasing or decreasing the length of a composite measurement.

The effect of a composite's length on reliability presumes that all the items that make up the composite score measure the same thing: in other words, that all the items are equally reliable measures of whatever the composite score measures. This imposes a practical limit on how long

The effect of a composite's length on reliability presumes that all the items that make up the composite score measure the same thing: in other words, that all the items are equally reliable measures of whatever the composite score measures. This imposes a practical limit on how long a composite measure can be made. For example, if the composite is a knowledge test, the limit will be the number of knowledge items that one can devise without straying from the knowledge domain being measured. The time available for administering the test will also limit the practical length of the test. However, within these practical limits, putting as many items in the composite as possible is an effective way to reduce the composite score's measurement error.

The standard error of a composite score is the standard error of an individual score, not a group of individuals' scores. However, the standard error of a score is a component of the standard error of the group mean. This comes from the fact that the variance of the raw composite scores is the sum of the variances of what are usually called the true scores, and the error variance corresponding to the reliability of the composite measure. Thus:

$$S_x^2 = S_t^2 + S_e^2 \qquad 5$$

where s_x^2 is the variance of the raw composite scores, s_t^2 is the variance of the true scores, and s_e^2 is the error variance owing to the reliability of the composite score. The variance of the true scores can be thought of as variance owing to individual differences among the individuals who generated the scores. For example, if the composite scores were scores on a knowledge test, the true score variance would represent individual differences in knowledge about the subject matter of the test.

The confidence interval for the mean of a set of composite scores is given by modifying the earlier Equation 2 to yield the standard deviation of the mean with separate components for true score and error variances. Thus:

$$S_{\bar{x}} = \sqrt{\frac{S_T^2}{N} + \frac{S_e^2}{N}} \qquad 6$$

Equation 6 shows that the confidence interval for the mean of a set of composite scores, that is, $\bar{x} \pm 1.96\ s_{\bar{x}}$, can be significantly, sometimes heavily, influenced by the reliability of the composite score. If the composite measure is well constructed so that its reliability is high, s_e can be become small enough to be virtually inconsequential. Now, the size of the confidence interval will be almost entirely a function of the individual differences among the individuals in the test sample. Conversely, if the composite measure is sloppily constructed and has poor reliability, the error owing to unreliability can make the confidence interval more a function of the composite score's unreliability than a function of the individual differences in the sample.

Ferguson [115] points out that measurement error is often ignored in behavioural experiments, and that consequently, the results of behavioural experiments are often effects that are largely from measurement error rather than from experimental treatments. Thus, minimizing

Coming back to the problem of sample design, the argument leading up to Equation 6 shows that while the standard error of a composite measure can be broken down into two components, the definition of the standard error for a composite measure remains parallel with the definition for a single-variable measure (see Equations 1 and 2). Consequently, all the principles that apply to sample size determination for single-variable measures also apply to composite measures. The only difference is that a composite measure allows a considerable degree of control over the measure's inherent error of measurement.

The error of measurement of a single-variable measure cannot be controlled as easily as it can for a composite measure. Nevertheless, the idea of a composite's error of measurement can be used to find ways to reduce error of measurement in single-variable measures. Thus, from Equation 6, the variance of the mean of single-variable measurement in a test design includes a variance component owing to individual differences among test participants, and a variance component owing to measurement error. In a single-variable measure such as test participants' accuracy of performance of a task, the error of measurement includes error caused by observers failing to record data correctly, observers being inadequately trained, and so on. This shows that procedures for constructing single-variable measures can have the same kinds of impacts on measurement error and sample design as procedures for constructing composite measures do.

4.4.5.6.3 Optimizing the Overall Design

The methods for optimizing a sample all apply to individual measures. Thus for example, the optimum participant sample size for one test measure will not likely be the same as the optimum participant sample size for another measure. This creates another optimizing problem: finding an optimum sampling plan that makes a reasonable compromise among the varying sample size requirements of the different test measurements. Probably the best strategy for doing this is to first rank the test measures according to each one's importance for the decisions that will be made from the test results. Then let the sample size requirements be determined primarily by the sample requirements for these test measures.

This strategy for finding the best compromise also suggests that the most important test measures should be identified ahead of time so that their sample size requirements can be worked out first. Determining an optimum sample for a variable can take much effort. In a design test with many different types of test measures, working out an optimum sample for each measure will not be practical. Therefore, concentrating optimizing efforts on the most important test measures will ensure that they are not missed out if being selective in deciding which test measures to optimize is necessary.

4.4.5.6.4 Reconciling Sampling Demands with Project Resources

The preceding on optimizing sample size said little about how project-specific resource constraints should be taken into account in constructing a design test. Project constraints include such things as budget, scheduling demands, availability of test participants, and so on. Project

constraints act as a set of boundaries within which a design test must operate. Every resource constraint, no matter what its nature, acts as a constraint on the possible sample size of one or more test measurements. This being so, it is always possible to construct a design test by taking the resource constraints and translating them into a sampling plan. Thus, the number of participants available for the design test becomes the participant sample size; the number of days available for running the test becomes the sample size for measurement replications, and so on. Although this is a temptingly easy way to define a sample, the resulting sample will not necessarily be optimum. Worse, it may not even be sufficient to answer the design questions to be asked in the test.

Project constraints typically represent a rough estimate of the design testing resource requirements. Because the estimate is made a long time ahead of when design tests will be conducted, and before much is known about what is to be tested, there is little reason to expect that this estimate will turn out to represent an optimum sample design. Thus designing a sample based on project constraints ignores any of the sample design information that will almost certainly be acquired over the course of the project. Failure to use this information to construct the test sample risks either of two unsatisfactory outcomes:

1 measurement samples too small to yield the statistical confidence levels needed for the decisions that have to be made from the test results, or

2 larger than necessary sample sizes that represent wasted resources that could have been put to better use.

Obviously, making a design test vulnerable to either of these two outcomes is inconsistent with a major intent of the CCIS HFE Requirements. It is also inconsistent with sound applied research practice.

Working out an optimum sample not only prevents waste of resources, but also give test results users one of their proper roles in constructing the design test. When the optimum sample shows that available resources will support the test, users of the test results can be given some assurance that the design test will produce useful information. If the optimum sample calculation shows that available resources will not support the sample, the calculation will still provide a rational basis for finding a compromise that balances resources with users' decision making needs. The appropriate compromise will depend on the specific circumstances of the design test. However, no matter what the available compromises, the people best qualified to decide which compromise to make are the users of the test results. While the HFE specialist can and should help the users make an informed choice of compromise, only the users can set their decision making priorities against what can be learned from the design test. In contrast, failing to attempt to calculate an optimum sample prevents users from exercising their rights and responsibilities as design test customers.

4.4.5.6.5 Summary of Design Test Experimental Design

1 The aim of a design test sample design is to optimize measurement sample sizes to balance how much confidence can be placed in the test results with the cost of the design test.

2 Optimizing a test design consists of optimizing the impacts of four sets of variables:

2.1 the errors of measurement of the test measures,

2.2 the test measurement sample sizes,

2.3 prior expectations about the likely results of the test measures, and

2.4 the gains and losses associated with the decisions that will be made from the test results.

3 The HFE specialist's primary means of minimizing sample size requirements lies in selecting and constructing test measures to minimize their errors of measurement.

4 Where compromises have to be made among the sample sizes of different test measures, the test measures that are most important to decision making should be given highest priority.

5 Optimizing the design test sample provides an important means of determining if resources available for the design test are adequate. When resources are not adequate, the optimum sample provides a rational basis for balancing design test priorities with resources.

4.4.5.7 Test Operations Scenarios

The test operations scenarios are the simulated operations that test participants will perform during the test. Operations scenarios will be defined by the test measurements to be taken, the test medium, and the sample design. Two types of scenarios can be used in a design test:

1 a set of exercises in which test participants do preassigned system operating tasks, and

2 a headquarters exercise (what an army calls a command post exercise or CPX) in which the test participants use the design system to carry out the tasks in all or part of a command and control cycle.

The first type of scenario would normally be used in a circumscribed test of particular system functions, or in a test of a preliminary design. The second type of scenario would normally be used in a final system acceptance trial, or in a test of an advanced design.

A scenario defines all the activities that the participants will do during the test. A scenario has three components:

1 the instructions that test administration personnel give to test participants to tell the participants what tasks to do on the system during different test exercises,

2 instructions to test administration personnel for initializing and running the scenario for each test exercise, and

3 for a headquarters exercise, the sequences of instructions specifying the tasks that the exercise controllers are to do in different exercises.

For a basic design test, scenario exercises can be constructed from user function and task descriptions. The exercises are constructed to provide opportunities to administer the various test

measures defined for the test. The design test sampling plan will determine how often the different exercises are to be administered. If learning effects from doing each exercise are likely, the sampling plan should also balance the sequence of exercises to compensate for the learning effects.

Once the scenario exercises have been identified, it is usually easy to translate the exercises into instructions for the test participants and test administrators. As recommended before concerning constructing measurement instruments, all the test administration instructions for both test participants and test administrators should be as written instructions. Besides ensuring that scenarios are properly administered, creating detailed written instructions helps in visualizing the scenario activities, and thus helps to identify and correct potential problems in administering the scenarios.

Scenarios for headquarters command and control exercises can usually be constructed by adapting one or more of the user organization's headquarters exercise scenarios for use as the design test scenario. The headquarters exercise, or CPX, is a primary training medium in all command and control organizations. Thus, the user organization will have almost ready-made scenarios. In a headquarters exercise, the exercise participants operate in a real or simulated headquarters facility.

Scenario events are simulated by means of the contents of messages being sent to the exercise headquarters, and by the contents of messages that the headquarters sends out in response to the scenario events. The exercise controllers play the roles of headquarters above and below the exercise headquarters. This use of controllers was described in the earlier example of a design test exercise (see Section 4.4.2, page 186). Figure 29 illustrates this concept. Controllers can play their roles in two basic ways: 1) by literally acting as commanders or liaison officers coming into the exercise headquarters to deliver orders, receive briefings, and so on, or 2) by sending and receiving messages over the command and control communications network. Since a headquarters is essentially an isolated facility, the headquarters' view of combat operations is a view derived from the message traffic it receives from other headquarters. Therefore, the realism of an exercise is entirely a matter of the degree to which this message traffic reflects the events in a real operation.

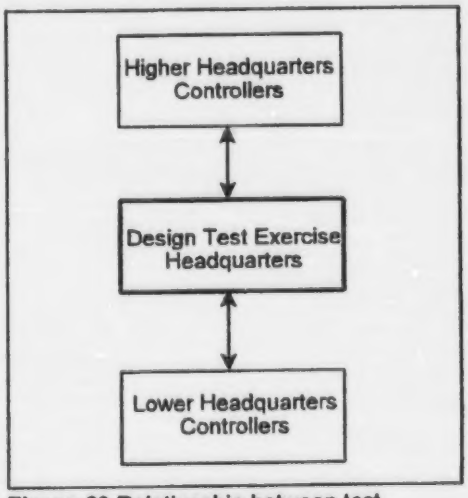

Figure 29 Relationship between test exercise headquarters and controllers.

In a headquarters exercise, controllers generally work from scripts that specify each message to be sent to the exercise headquarters, and the time when the message is to be sent. For example, the first message starting a scenario could be the orders from higher headquarters to begin a command and control cycle. Messages sent by the exercise headquarters are normally logged the same way as they would in real operations. Thus, the scenario scripts and the message logs provide a complete record of the play

during the exercise. Often, modified existing exercise scenarios can be used for design tests. Usually, the only modifications needed are ones to provide enough test measurement opportunities to meet the requirements of the sampling plan.

Although having ready-made scenarios is a great convenience, it still requires a good deal of careful effort to set up scenarios so that they will meet the test measurement requirements. Most of this effort will go to ensuring that the data collection procedures will be synchronized with the scenario events. The most effective organization for running a headquarters exercise is to make it a user organization responsibility, with the system development contractor being responsible for the test measurement provisions. Normally, only the user organization has the personnel with the experience. Knowledge needed to set up and operate an exercise. Therefore, putting the exercise primarily in the hands of the user organization is the most practical way to ensure that it will serve the aims of the design test.

4.4.5.8 Develop Participant Training

For the test participants' performance on test measures to represent how users would eventually perform on the system, test participants have to be trained to operate the test design. The training requirements can be derived from the user function and task descriptions, and the test scenarios. Generally, developing training for the test should be a straightforward translation of user task information into a training curriculum.

Time for training test participants is sometimes viewed by HFE personnel as a needless overhead cost. Because of this, the CCIS HFE Requirements User Manual (4.3.8) devotes significant space to describing why participants need to be properly trained for doing the exercise tasks in a design test, and listing the project benefits that can be gained from training for design tests. The benefits are the following.

1 The ease or difficulty with which participants learn to operate a design is itself an indication of the HFE merits of the design.

2 Training sessions provide opportunities for practising and refining data collection procedures, and an opportunity for test participants to get oriented to the data collection activities.

3 Training participants for design tests provides information useful for setting up training when the system is implemented.

The only test design technical requirement that needs to be met in participant training is to ensure that all the participants are trained to a common performance criterion. The training should include performance tests that each participant must pass before the design test activities start. If participants' levels of training success are unknown when the design testing starts, it will be very difficult to partial out training or learning effects in subsequent data analysis. If it is practical, baseline data on some test measurements should be collected at the end of training. In the later data analysis, these baseline data can be used as co-variates to partial out residual learning effects, or individual differences among participants' training successes.

Requiring participants to meet a performance criterion before completion of training implies that some provision may need to be made for giving remedial training to participants who have difficulty learning the system. Having a training criterion also suggests that the criterion should not be too stringent. Making the training too burdensome will risk discouraging the participants before they start the test exercises, and possibly risk loss of their cooperation. The training criterion only needs to be sufficient to ensure that participants have a reasonable level of system operating skill before design testing starts. If there are remaining learning effects, they can usually be taken into account in the data analysis.

Finally, if a series of design tests is planned, and training is a substantial cost, using the same participants in successive tests may be necessary. Having participants from past tests reduces the time needed for training in subsequent tests. The disadvantage of this is that it reduces the representativeness of the overall sample of people who participated in the tests. It similarly reduces sample size and consequently increases measurement error.

4.4.5.9 Pilot Test

The purpose of a pilot test is to make sure that all components of the design test will function as planned. If the design test represents a substantial investment of resources, a pilot test should be considered an absolute necessity for avoiding losses from a design test that does not work. The design test expenditures that could be wasted, because no pilot test was done, include

1 the personnel costs associated with the test including:
   - 1.1 user organization personnel serving as test participants,
   - 1.2 user organization personnel assisting in the design test, for example by serving as exercise controllers,
   - 1.3 contractor personnel conducting the test and providing system operating support,
   - 1.4 any designer downtime resulting from delays in getting test results, and
   - 1.5 contractor and acquisition authority personnel providing administrative support for the test,
2 the test facilities and equipment costs including,
   - 2.1 the operating cost of the design test medium, and
   - 2.2 the cost of the physical facilities used for the test, and
3 any indirect project costs arising from failure to complete the design test successfully and on time.

Considering all these factors, even a modest design test represents a considerable investment. Moreover, the more expensive a design test is, the more complex will the test usually be, and the more things will there be to go wrong. Thus, the more expensive the design test, the more critical it is to do a pilot test.

A pilot test can be used to test the following components of a design test.

1 *Test measurement properties:* get estimates of the errors of measurement of the test measurements to verify that the sample sizes in test design are adequate.

2 *Test medium:* verify that the test medium accurately simulates the design features to be tested, that it can support all the test measurements and operations scenarios, and will function without major breakdowns during the test exercises.

3 *Operations scenarios:* confirm that the operations scenarios are realistic, will function properly, and that the test measurement procedures will not unduly interfere with participants' performance in the scenarios.

4 *Participant training:* confirm that the participant training procedure will make participants proficient enough to allow treating the performance measurement data as data from reasonably competent system users

5 *Test administration personnel and procedures:* confirm that all test administration personnel know their duties, and that test measurement administration procedures work as planned.

6 *Test Schedule:* determine if the times allowed for different test activities will be sufficient for collecting all the test measurement data.

It is sometimes possible to pilot test these components one or two at a time. For example, test measurement administration procedures can be tested in a mock design test. However, if the design test is at all complex, running the design test as a full dress rehearsal for the actual design test is the safest thing to do, particularly for the first design test. If a series of design tests is being conducted, experience from previous tests will reduce the need to do extensive pilot testing.

When a design is tested, particularly for the first time, it is often the first time that personnel from the user organization will get to see the system in operation. From the contractor's point of view, and especially from the HFE specialists' point of view, it is very important that the design test activities run smoothly. If the test activities do not run smoothly, user organization personnel and acquisition authority personnel will inevitably suffer a loss of confidence in the design system and the contractor. When a design test does not run smoothly, test participants and other observers can seldom sort out whether the design being tested was faulty or whether the testing procedure was faulty. Generally, a poorly run design test reflects badly on the design no matter how good the design may in fact be. The only way to ensure that a design test shows the true merits of the design is to pilot test the design test procedures thoroughly.

4.4.5.10 Select Test Participants

To represent the population of potential system users, design test participants must be a random selection of individuals from the population of people who will be the eventual users of the system. This is seldom possible in practice because the user organization cannot make all its prospective users contingently available for the design test. A pool of potential users from which test participants could be randomly selected would be the next best thing to having the entire potential user population available. In practice, even this second-best option may be difficult to

achieve because identifying a pool of available personnel can be just as difficult as making everyone available. This will leave the test designer with no option but to ask the PM to recruit whoever can be found to serve as test participants.

If a random selection is not possible, test participants should be selected to be as representative as possible in the circumstances, and simultaneously, be selected to be free of biases that could influence the test results. This can be achieved with participants who:

1 represent a cross-section of user seniority and experience;

2 are not day-to-day coworkers;

3 come from different units of the user organization;

4 have not had any role in the system acquisition or development process other than acting as test participants, and

5 have not been in any way hand-picked for any prejudices they might have, or be thought to have, about the design being tested.

The first three criteria help to achieve representativeness. The second and third criteria guard against the temptation to recruit all the participants from one or two units that might happen to be available during the design test. Because each unit in an organization typically develops a certain degree of uniqueness in outlook and operating procedures, selecting all participants from the same unit or team limits the representativeness of the sample.

Screening out candidate participants who have had a role in development, or who might have a bias for or against the new system, is a precaution against getting test participants who could bias the test results in a particular direction. Candidates who have participated in any previous design tests should be treated as candidates who have had a role in the system development.

In the fifth criterion, screening out candidates who might be hand-picked, the emphasis should be on the term, hand-picked. Chances are that any pool of candidates will have candidates with prejudices for or against the system. For a sample to be random, a prejudiced candidate should have the same chance of being recruited as any other candidate. In contrast, hand-picked participants had a higher chance of being recruited than other candidates did. They therefore represent a sampling bias that reduces the validity of the design test.

When the drawing of a sample is not strictly random, two effects need to be considered. One is the effect on test measurement variances. The other is the effect on test measurement bias. The variance of any test measurement will include a component owing to sampling variance. Nonrandom sampling usually makes the sampling variance smaller than it would be with random sampling. However, in certain situations, nonrandom sampling can make the sampling variance larger than it would be otherwise. Sudman [112] and Kish [114] describe how sampling variance can be affected by different sampling procedures. Although there are ways to estimate the size of a sampling variance effect, they are seldom practical to apply in a design test. Therefore, the best way to deal with sampling variance effects is to avoid sampling procedures that would have any extraordinary effects on sampling variance.

The biases of most concern in a design test are attitudes for or against either the particular system being tested, or for or against the system's technology. These biases are usually easy to detect when participants start doing the test activities. The biases are not easy to measure. Biases will most likely show up as response bias effects in opinion measures. As discussed earlier in Section 4.4.4.2.5.3, the only real way to protect against response bias in opinion measures is to avoid using opinion measures without having complementary user performance measures.

4.4.5.11 Test Schedule

The test schedule specifies the dates and times when all the test events will occur, including:

1 when participants will join and take part in the test;

2 completion of the development of test measurements and test media;

3 pilot tests of test measurements and test media (including time needed to revise);

4 the design test itself, and

5 test results data analysis and reporting.

Constructing and managing the test schedule requires cooperation between the HFE personnel running the design test and the PM. Overall, the PM will be concerned with how the timing of different test events will affect overall project scheduling. The CCIS HFE Requirements User Manual (4.3.11) discusses the PM's test scheduling concerns. The following will deal with scheduling matters that the HFE team has control over.

HFE personnel control how long it will take to develop the test measures and test medium, conduct the pilot test, conduct the design test, and produce the results of the test. HFE personnel control these activities by virtue of the fact that how long it takes to do each activity depends on how much work that has to be done to complete the activity. Since HFE personnel control the duration of each activity, HFE personnel are also responsible for providing accurate estimates of how long each activity will take when the test schedule is made up. HFE personnel are further responsible for ensuring that each activity is completed according to the schedule.

The accuracy of an HFE specialist's prediction of how long an HFE activity will take generally depends on the specialist's experience with doing the activity. If one has limited experience with a task, there is a tendency to underestimate how long the task will take. Therefore, allowing extra time for work that is new or work that contains unknown factors is wise. In cases where the work is familiar enough to be almost routine, scheduling predictions can be quite accurate.

Although projects' schedules always create pressure to hurry work along, nothing is gained by yielding to time pressures at the cost of doing shoddy work. Therefore, laying claim to as much time as possible for each activity is the best way to avoid being pressured into doing work that is less than complete to one's own satisfaction.

4.4.5.12 Provisions for Conducting the Test

Section 4.4.5.9 (page 242) included a discussion of the importance of a pilot test to ensure that all the technical components of the design test functioned smoothly. Having the test run smoothly is important not only for technical reasons, but also for maintaining the acquisition authority's confidence in the design project. If the acquisition authority loses confidence in the project, it should be for reasons connected with the merits of the design, not because the HFE team was unable to mount a well-run design test.

Although having sound technical components are fundamental to a well-run test, additional provisions need to be made to support the technical components of the test. These provisions are things that should be done to ensure the comfort of the test participants and maintain their cooperation during the test exercises. Table 21, (adapted from [17]) lists the things to do to ensure participants' cooperation during the testing activities. This table is also in the CCIS HFE Requirements User Manual.

Table 21 Things to do to gain participants' cooperation.

1	Make sure the participants know that their participation is important for the evaluation of the system.
2	Provide each participant with a written guarantee that individual data will be confidential to personnel directly responsible for analysing the test results.
3	Before and during the test, keep participants informed about what they will be doing in the test, and the purpose of each test activity.
4	Treat the participants as experts on the operational environment where the system will be installed.
5	Emphasize that the participants are testing the system - the system is not testing the participants.
6	Treat the participants as guests, and make sure that they have the same rest facilities and privileges as any other test personnel.
7	Encourage comments and criticisms. Provide a comments log where participants can write comments not covered by interviews or questionnaires.
8	Avoid defending the system when participants point out system shortcomings.
9	When participants have difficulties learning or performing operating tasks, avoid suggesting or implying that the difficulties are due to the participants' shortcomings.
10	Keep non-essential personnel out of the test area.

Participating in a system design test imposes a certain amount of stress on test participants. When test participants are subjected to extraneous stress, they cannot perform as well as they would otherwise. This will act as a bias in the test measurements - usually a bias unfavourable to the system design. All the provisions in Table 21 are things to do to reduce this stress and its consequences for the results of the design test. The provisions in Table 21 are also similar to advice given by Nielsen [46] for treatment of participants in software usability tests.

Confidentiality of test results is an important general ethical principle that applies to all research with human subjects. Individual test results should be available only to individuals directly concerned with analysing the data. The contract or agreement under which the test is performed

should include a clause affirming the confidentiality of the test data. To protect confidentiality, raw test data such as completed questionnaires should be treated as confidential documents. Participants should be identified in computer files only by alphanumeric identification codes. Identification codes can be issued directly to participants for identifying themselves on questionnaires and other forms. This can be done in a way that prevents even the testing staff from being able to identify individual results. Before using this form of protection, several factors should be considered.

1 Contacting participants after the test to clarify data or verify illegible or missing responses may not be possible.

2 Participants can easily forget or incorrectly record their identifiers; something they are unlikely to do if they identify themselves by name. If identifiers are recorded incorrectly, associating the misidentified data with the correct participant may be difficult later.

3 It will be impossible for a third party to verify the provenance of the data in case of a dispute, or in case the contract has a provision for a third party audit or verification of the test data.

4 If the person directly responsible for the test has a professional license or certification, there is often a professional rule requiring that investigators doing research on humans retain the names of research participants and be able to associate data with individual participants. This is notwithstanding the requirement also to protect individual confidentiality.

Related to confidentiality requirements, general ethical rules for human research also require that participants give their informed consent to participate in the research, and for their data to be used for the specified purposes of the research. The person directly responsible for conducting the test should make himself or herself aware of any laws or regulations that might apply to the human research aspects of the testing activities.

Possibly with one or two exceptions, the items in Table 21 are easy to carry out and do not cost much. The potential exceptions are items 7 and 8. People even remotely connected with the system development, including not only contractor personnel but also acquisition authority personnel, often find it difficult not to defend the system design when test participants point out system shortcomings. Whether a participant's opinion is valid or not, the participant must never be discouraged from commenting on system design features. When a participant comments on the system, the only allowable response is either a neutral recognition of the participant's view, or a suggestion to the participant to tell the design test administrator. Often the problem of maladroit responses to participants' comments arises not in the controlled testing environment, but in social contacts during rest breaks, lunch periods, and the like. It is in these informal contacts that project personnel can unwittingly get themselves into the position of explaining away perceived system deficiencies. Where there are likely to be such contacts, all project personnel who might meet the test participants should be asked to avoid doing anything that discourages participants from freely expressing their thoughts.

The comments log in item 6 is a good way to give participants an outlet for spontaneous reactions to the system design. The contents of the log can then be discussed in debriefing sessions. Although not all the suggestions need responses, an exception should be made where the participant clearly shows a misunderstanding of the system's function or purpose. Participants

should not return home after the test sessions with serious misunderstandings about the system design.

In general, making participants feel comfortable in the design test situation, and valuable to the development effort, will generate cooperation and contribute to producing a smooth well run design test.

4.4.5.13 Results Analysis and Reporting

The CCIS HFE Requirements DID titled Human Factors Engineering Test Report provides a detailed outline for a design test report. Complying with all the requirements in this DID should ensure a comprehensive analysis of the test results and an equally comprehensive set of prescriptive design recommendations. Although this DID provides a good basis for data analysis and reporting, a few items may need special attention. The following deals with these items.

4.4.5.13.1 Interpretation of Statistical Confidence Intervals

The use of 95 and 99 percent confidence intervals as statistical significance criteria is so pervasive in scientific literature that is often forgotten that these confidence intervals are purely arbitrary conventions for judging whether or not an experimental result is 'significant.' In applied research, a result's decision making implications are much more important than whether the result is statistically significant. This outlook is consistent with that described in Section 4.4.5.6 (page 230) for determining a sample size. Thus, the confidence interval a decision maker requires may be substantially less than 95 or 99 percent, or it may be considerably greater than 95 or 99 percent. This implies that data should be analysed by reference to the decision maker's payoff matrix rather than by blindly sorting results into categories of statistically significant and statistically not significant. In design testing, the statistical hypothesis testing problem is to decide whether a result is material to recommendations about the system design, not to find out whether a result is statistically significant.

4.4.5.13.2 Analysis of Rating Scale Data

Since rating scales are used so widely to get test participants' opinions about design issues, it is worth mentioning a common problem in the analysis of rating scale data. The first step in these data should be to carefully to inspect the distribution of the ratings on each rating item. If the distributions on many items are highly skewed so that they bunch up at one or the other ends of the scales, any analyses that assume normally distributed data can produce misleading conclusions. If many items have prominently skewed distributions, transform the data to a scale with fewer rating intervals, or transform the data to a dichotomous scale. Although transforming the data this way causes some loss of information, it considerably reduces the chances of drawing misleading conclusions from the analyses.

4.4.5.13.3 Analysis of Data From Composite Test Measures

Data from composite measure such as knowledge tests or aggregated rating scales should be analysed to calculate the reliability and standard error of each measure. If a measure is reliable, it can provide useful information about individual differences in the test participant sample that cannot be obtained from single-variable measures. Using item analysis, it may also be possible to increase the reliability of the composite measure by eliminating items that do not correlate with other items in the composite.

If a composite measure and a single-variable measure are correlated, analyses can sometimes show how much of the single-variable's variance was from individual differences in the participant sample, and how much was from measurement error. Having an estimate of the variance from individual differences amounts to increasing the precision of the test measurements. A better way to get at variance owing to individual differences is to have an experimental design that permits an analysis of variance in which the between-subjects variance can be calculated. However, this kind of design is seldom practical in a design test.

4.4.5.13.4 Timely Reporting

A report of a substantial design test based on the pattern Human Factors Engineering Test Report DID requires a good deal of preparation time. This preparation time may delay system development decisions that depend on the design test results. A way to avoid this delay is to plan the test reporting so that a preliminary report can be completed as soon as the main test data are analysed. This would allow other work to continue while the full report was being completed. Issuing a preliminary report should not be a substitute for a full report. To ensure this, the wise PM will demand completion of the full report before the next project review meeting.

Finally, the HFE Test report is a fundamental component of the documentation of the HFE work performed in the development of the system. If the project is an intermediate development phase, the HFE test reports will be the foundation for design test work in future development phases. If the test report is from a final development phase, it will be the foundation for a variety of post implementation and system life cycle activities. Therefore, the HFE design test report deserves the HFE specialist's best effort.

4.4.5.14 Summary of HFE in Design Testing

The HFE activities in conducting a design test can be summarized as the performance of the following tasks.

1 *HFE Test Plan.* Construct a test plan describing how the test will be developed and executed.

2 *Purpose of the Test.* State the purpose of the test including specific system design questions to be answered by the results of the test.

3 *Define Test Measures.* Identify candidate test measures from the system design questions and the user function and task analysis data. For each test measurement: 1) identify a measurement metric, 2) specify a measurement criterion value indicating adequate user or system performance, and 3) estimate the measurement's validity and reliability properties. Select the set of measurements that provide the most cost-effective evaluation of the design.

4 *Test Medium.* Specify the design test medium capable of representing the system design and supporting the test measurements.

5 *Define Operations Scenarios.* Translate the user function and task descriptions into operations scenarios that will support the test measurements.

6 *Participant Training.* Translate the user function and task descriptions into participant training covering the operations scenarios to be used in the test.

7 *Test Statistical Design.* Specify the participant and test measurement samples based on: 1) the test measurement criteria defining adequate user and system performance, 2) the statistical criteria defining confidence in the validity or accuracy of the test results, and 3) the estimated statistical variances of the test measurements.

8 *Pilot Test.* Conduct a pilot test of the test design to verify the test measurements, test administration and data collection procedures, test medium, test scenarios, and participant training.

9 *Test Participants.* Within the practical limitations of the development situation, define participant selection criteria that will yield a sample of participants who will represent the system's eventual users, and be unbiased in their attitudes to the system design.

10 *Test Schedule.* Construct a test schedule for all the significant activities in the development and execution of the test. Confirm that all parties involved in the test can meet the schedule's timings.

11 *Provisions for Conducting the Test.* Make any provisions necessary to ensure that participants will have comfortable facilities for doing the test activities, that they will be treated with tact, and they will be shown appreciation for their participation in the test.

12 *Results Analysis and Reporting.* Describe how the test was done in sufficient detail for report readers and any third parties to evaluate the technical soundness of the test. Analyse the test data according to the system effectiveness and statistical criteria defined earlier. Formulate the results of the analyses into recommendations for the system design.

The final design test in the system development plan will mark the completion of the HFE development and testing activities, and the transition to system implementation activities. The next section describes HFE activities in system implementation.

4.5 HFE in Implementation

HFE in implementation consists of HFE activities to support the implementation of the new system. The CCIS HFE Requirements does not contain any requirements for HFE in

implementation. However, there are several HFE activities that can contribute to implementing a new system in operations, and the CCIS HFE Requirements can be useful for planning and managing these activities. Possible HFE activities in implementation include the following.

1 Give the user organization comprehensive documentation of the final version of the user function and task analysis results.

2 Develop a user training curriculum from the user function and task information.

3 Develop additional job aids for training and operations.

4 Conduct the user training program.

5 Develop and conduct a train-the-trainers course for training the user organization's training staff to train users.

6 Define training success criteria and tests of training achievement.

7 Identify personnel selection criteria for selecting people who will operate the system.

8 Provide data for personnel functions such as job classification, evaluation, and analyses of staffing requirements.

9 Conduct post implementation HFE reviews or tests.

All these activities are applications of the HFE analyses that describe users' tasks in the final version of the system. If the development project includes implementation services, the implementation HFE activities can be specified by adapting sections of the CCIS HFE Requirements and its accompanying DIDs.

4.6 Summary of HFE Methods

The HFE activities in CCIS development divide into four categories roughly corresponding to the different phases of the system development plan.

1 *Identify system requirements.* Do mission, function and task analyses of users' operations to identify system requirements for supporting users.

2 *System design.* Provide direction on the design of the system-user interface and the system's facilities for supporting users' tasks. Also, maintain descriptions of users' functions and tasks in the emerging system design.

3 *Design testing.* Translate user function and task information into measures of user performance, design and conduct user tests, and analyse test results to identify system design revisions.

4 *Implementation.* Translate user function and task information into user training support and help the user organization in setting up user training.

Table 22 shows a summary of all the HFE tasks in a CCIS development plan. This past section has described the technical content of the HFE activities that go into developing a CCIS. The next section describes HFE responsibilities in managing these activities.

Table 22 Summary of HFE tasks in system development.

Development Phase	HFE Tasks
System Requirements	- HFE analyses to identify deficiencies or define system requirements. - Human-machine function allocation.

System Design	- Identify applicable HFE standards and guidelines.
	- Translate HFE analysis data into design requirements.
	- Keep HFE analyses current with the system design.
	- Review HFE merits of design proposals for:
	  - compliance with HFE standards and guidelines,
	  - achievement of the system functionality requirements, and consistency with human capabilities,
	  - effectiveness of user interface,
	  - estimation of user workloads,
	  - identification of measures of user performance and system effectiveness,
	  - identification of user skills and knowledge requirements,
	  - identification of training requirements,
	  - evaluation of standard operating procedures.
	- Conduct HFE design tests.

Design Testing	- Construct HFE Test Plan. - Identify design questions to be answered by the test. - Identify test measures including each measurement's metric, user or system performance criterion, and validity and reliability. - Specify design test medium for supporting test measurements. - Develop test operations scenarios. - Develop participant training for test scenarios. - Develop optimum test statistical design. - Conduct pilot test of design test. - Define participant selection criteria. - Construct a test schedule for all the significant test activities. - Identify measures for ensuring participant cooperation. - Conduct test, analyse data and report findings for prescribing design improvements.
Implementation	- Provide documentation of user function and task analysis results. - Develop a user training curriculum. - Develop job aids for training and operations. - Conduct user training, or train-the-trainers course. - Define training success criteria and tests of training achievement. - Identify personnel selection criteria for selecting users. - Provide data for personnel management functions. - Conduct post-implementation HFE reviews or tests.

5 DEVELOPING AND MANAGING THE HFE PLAN

This section provides advice to HFE specialists on creating and managing an HFE plan. The advice in this section is a counterpart to the advice on HFE project management provided for PMs and PMO staff in CCIS HFE Requirements User Manual (5). The User Manual contains a good deal of information that should be useful to review as part of developing an HFE plan.

Advice on creating and managing the HFE plan is in two parts: 1) constructing an HFE plan, and 2) managing the execution of the HFE plan.

5.1 Constructing an HFE Plan

The following describes the factors that need to be taken into account in constructing an HFE plan that will fulfill the requirements in the CCIS HFE Requirements. Constructing an HFE plan consists of doing two sets of tasks.

1 Identify the project technical and management requirements that the plan must satisfy, including:

  1.1 HFE requirements specified or implied in the project RFP and SOW,

  1.2 requirements defined by the system development plan.

2 Construct the HFE plan, including:

  2.1 construction of the technical plan,

  2.2 provision for coordination with the system development plan, and

  2.3 HFE plan budget and schedule.

5.1.1 Identification of HFE Plan Requirements

The CCIS HFE Requirements is intentionally designed to make it easy for the PM to specify HFE requirements for a CCIS project. CCIS HFE Requirements User Manual (5.3.2.2) describes options for specifying the HFE requirements in an SOW. Briefly, the CCIS HFE Requirements User Manual suggests two basic options. The simplest option is to invoke the CCIS HFE Requirements with a statement like the following.

> 'The system development plan shall include a human factors engineering plan (HFE Plan) conducted in accordance with *Guidelines for Human Factors Engineering Requirements for Canadian Forces Command and Control Information Systems* (CCIS HFE Requirements)'

The PM's second option for stating the HFE requirements is to elaborate this statement by adding tailoring specifications. CCIS HFE Requirements User Manual (5.3.2.2) outlines ways to add tailoring specifications for different types of projects. Then the PM can specify which paragraphs of the CCIS HFE Requirements are to apply to the project. Alternatively, the PM can just add a sentence to the basic statement of requirements to underline the tailoring requirement in the CCIS HFE Requirements (6): thus,

> 'The system development plan shall include a human factors engineering plan (HFE Plan) conducted in accordance with *Guidelines for Human Factors Engineering Requirements for Canadian Forces Command and Control Information Systems* (CCIS HFE

Requirements). The HFE Plan shall be tailored to activities in the overall system engineering plan in accordance with paragraph 6 of the CCIS HFE Requirements'

Since the CCIS HFE Requirements explicitly specifies that the HFE requirements are to be tailored to the nature and scope of the project, the additional mention of Paragraph 6 is unnecessary.

If the SOW only invokes the CCIS HFE Requirements without any tailoring specifications, it will then be up to the HFE planner to do the necessary tailoring. Then, referring to the CCIS HFE Requirements User Manual (5.3.2.2) will provide guidance on how to do the tailoring. the CCIS HFE Requirements User Manual (5.3.2.2) contains a series of tables identifying which parts of the CCIS HFE Requirements apply depending on whether the project is: 1) requirements study, 2) initial development project, 3) advanced development project, or 4) engineering development project. Table 1 (page 20) in this CCIS HFE Design Guidelines shows the HFE activities associated with these different types of projects. Tables 10 to 13 in the CCIS HFE Requirements User Manual identify the applicable CCIS HFE Requirements paragraphs for each type of project. These tables can be used to identify the applicable paragraphs if the project SOW does not identify them.

If the project is to include system implementation assistance, the SOW should contain a separate statement specifying the HFE requirements for implementation. This statement should specify the HFE tasks that are to be performed in implementation. The CCIS HFE Requirements User Manual (5.3.2.4) describes how to specify the implementation requirements.

5.1.2 HFE Planning Information Requirements

The system development plan HFE requirements are project-specific requirements that the HFE plan must meet. They include, not necessarily in this order:

1 system development plan requirements,

2 system design requirements,

3 requirements for level of effort background information, and

4 schedule and budget constraints.

5.1.2.1 System Development Plan Requirements

A system development plan is in many ways like a military operations plan, and the process of developing a system development plan can be much like the process shown earlier in Figure 2 (page 37) for developing an operations plan. For a commander, the critical ingredients for starting a military planning process are having a clear idea of the operations objectives, and a clear concept of the operation that would achieve these objectives. Having a concept of operations

quickly yields a plan concept or outline plan, and thence, a set of planning directions to the various specialists who will work out the details of the operations plan.

By analogy with the command and control planning process, the HFE planner will, ideally, start the HFE plan having in hand an outline of the overall system development plan, along with guidance on the project-specific requirements for the HFE plan. In practice, things may not happen this way. The HFE planner may be left for some time to his or her own devices before a system development plan begins to emerge, or before getting guidance on how HFE activities are to fit into the overall system development plan. Although the CCIS HFE Requirements and its DIDs define much of the HFE plan independent of a system development plan, the HFE planning should not be allowed to go far before getting planning guidance from the contract project manager. This guidance should include direction how best to support the system development effort.

5.1.2.2 HFE Plan and System Design Requirements

The system design requirements are the performance requirements that the new CCIS will be expected to achieve. As explained earlier in Section 3.2.1.2 (page 16), the system design requirements come from the user organization's statement of the deficiencies of its existing system. An ideal statement of a deficiencies implicitly defines the requirements that a new system would have to meet to remedy the present system's deficiencies, and defines objective performance criteria for verifying that the new system meets the requirements. The CCIS HFE Requirements User Manual (5.2) gives the PM guidance on developing a set of design requirements that defines explicit system performance requirements that the new system design must meet.

Templates for Statements of Requirements, adopted by the Directorate of Business Change Management include several sections that address human factors issues. These entries will not provide design solutions, but should identify major constraints and top-level requirements. Thus, much of the material needed to develop an HFE plan should be provided to the developer at the outset of a project and the HFE specialist should review this material carefully when developing the HFE Plan.

For the purposes of the HFE plan, the design requirements should define all the user performance requirements that the new system must support. These requirements will be specifications of speed and accuracy of user performance that the system is to support. These correspond to the types of performance and effectiveness measures described earlier in Section 4.2.7 (page 121). As Section 4.2.7 explains, the system design requirements associated with user performance and effectiveness are the future basis for conducting design reviews and design tests. They are therefore also a large part of the basis for planning design reviews and design tests.

If the system design requirements do not specify a satisfactory set of user performance requirements, only one possible course of action is possible: that is to begin either the project as a whole, or the HFE plan with a phase aimed at defining a set of explicit design requirements. If the

design requirements do not include an adequate set of user performance requirements, chances are that the design requirements for the system as a whole will also be inadequate. In this latter case, an initial requirements definition phase would be appropriate. If it is only user performance requirements that need to be better defined, then a separate user performance requirements definition phase should be put into the HFE plan.

Subsections of the CCIS HFE Requirements User Manual (5.2.3) describe procedures for analysing the user organization's statement of deficiencies to see if the statement is adequate for defining the system requirements. The Manual also gives procedures for improving the statement of deficiencies if the original statement turns out to be inadequate for defining design requirements. The subsections describe

1 (5.2.3.1) how to review a user organization's statement of CCIS deficiencies to see if the statement provides an adequate basis for developing a project statement of requirements;

2 (5.2.3.2) how to improve a weak statement of deficiencies to make it adequate for preparing a statement of requirements, and

3 (5.2.3.3) how to use HFE analyses to develop a new statement of deficiencies.

Doing the HFE analyses described in 5.2.3.3 would amount to doing an HFE analysis of users' functions and tasks in the predecessor system in sufficient depth to identify a comprehensive statement of deficiencies. Doing this would also perforce satisfy most or all of the requirements in the CCIS HFE Requirements (5.1).

From the perspective of design requirements, the initial HFE analyses required in the CCIS HFE Requirements can be viewed as simply a way to get a set of explicit user performance and effectiveness requirements. However, no matter what purpose is attributed to the initial HFE analyses, or whether they are even done, the HFE plan must ensure that the system design requirements include a set of user performance requirements before design begins. As the CCIS HFE Requirements User Manual (5.2.2) explains, not having explicit system design requirements creates all the project management problems that emerge when a project does not have clearly defined objectives. These problems include excessive concentration on project management activities at the expense of concentration on actual system development, excessive amounts of time spent on requirements analyses, and continuing uncertainty about whether or not the emerging system design will meet the users' needs.

5.1.2.3 Requirements for Level of Effort Information

Budgeting and scheduling for the HFE plan requires having background information needed to estimate how much work it will take to do each HFE activity. The effort that will be needed for the HFE plan depends on two things, 1) the scope of the HFE plan, and 2) the size and complexity of the user organization. The scope of the HFE plan is defined by which paragraphs of the CCIS HFE Requirements have been invoked for the project. These paragraphs specify the

HFE tasks that will be performed during the project. The size and complexity of the user organization generally determine how much effort that will go into each HFE task.

The effort needed to do the project HFE activities is almost entirely a function of the number of different user jobs (not the number of users). Nearly all the HFE tasks are affected by the number of different user jobs. For example, each job requires a separate task analysis. And the more jobs there are, the more complex will the system be, and the more system design functions will there be to develop, review and test. Additional factors that influence the level of effort include the complexity of the jobs and the amount of interaction among different jobs.

Since the number and nature of the user organization's jobs strongly influences the size of the HFE effort, information about the user organization's jobs and operations is indispensable for estimating the HFE effort. An organization chart and an overview of operations are minimum requirements for estimating the HFE level of effort. If at least this information is not available, there is no sound basis for estimating the HFE level of effort. The more additional information there is about the user organization's operations, the more accurately is it possible to estimate the HFE work requirements.

The estimate of the HFE analysis effort should include an estimate of how much information can be provided from other system analysis activities. System analysis data should provide the bulk of the HFE mission and function analysis data. Depending on the system analysis approach being taken, getting a good deal of task analysis information from the system analysis may be possible, or possible to combine task and system analysis work to share user task information.

In summary, estimating effort required for the HFE plan requires specification of the CCIS HFE Requirements paragraphs that will be applicable to the project, and, at a minimum, basic knowledge of the user organization's personnel organization and their operations.

5.1.2.4 Budget and Schedule Constraints

Obviously, schedule and budget constraints have to be taken into consideration in constructing the HFE plan. For budgeting, the CCIS HFE Requirements User Manual (5.3.2.5) suggests that five to 10 percent of the total system development budget is a reasonable expectation for the cost of the HFE effort. However, this figure should not be taken for granted. Instead, the HFE budget should be worked out based on a draft HFE plan. If the estimated cost of the plan is in the five to 10 percent range, it is probably reasonable. It is seldom a good idea to plan any project by putting the budget at the forefront of the planning considerations. It is better to first construct a technically sound plan independent of budget restrictions. If the plan comes out more expensive than the budget permits, the plan can be trimmed to fit the budget. Trimming a plan to fit a budget at least shows the technical sacrifices that will be made by trimming the plan. This in turn, provides a basis for trimming the plan in a way that will least weaken the technical soundness of the original plan. Creating the plan before worrying about the budget also provides a basis for seeing if the cost is higher than expected for good reasons. If the cost is higher than expected for good reasons, a good case can then be made for increasing the budget allocation.

For the HFE plan schedule, the only information needed is the overall system development schedule. To influence system design, various HFE activities must be completed at the times when their results will be needed as system design inputs. Most HFE activities can be done in parallel with other system engineering activities. The HFE activities usually also take about the same times to complete as corresponding system engineering activities. Therefore, synchronizing the HFE plan with the rest of the system development plan should be easy to do. The only potential synchronization problem occurs when design activities have to be delayed until design test results become available. This problem can usually be avoided if the contractor project manager takes care to coordinate design and testing activities to avoid unnecessary delays or unnecessary idle time.

5.1.2.5 Summary of HFE Planning Information Requirements

The information needed for constructing an HFE plan includes the following.

1 *System development plan information.* Information about the system development plan provides the basis for coordinating HFE activities with other system development activities. System development planning information should also provide guidance on how the HFE activities could best support the general system development process.

2 *System design requirements.* The system design requirements are the performance requirements that the new CCIS will be expected to achieve. Having objectively verifiable user performance and effectiveness requirements is critical to the success of HFE design and design testing activities. If an adequate set of design requirements is not available at the start of the project, the HFE plan should include provisions for developing an adequate statement of the design requirements.

3 *Requirements for level of effort background information.* The level of effort required for the HFE work depends on which CCIS HFE Requirements paragraphs are applicable to the project, and on the number of different jobs that will be affected by the new system. A description of the user organization's personnel organization and basic descriptions of the personnel functions represent minimum information for estimating the levels of effort required to do the activities in the HFE plan.

4 *Budget and schedule constraints.* Budget constraints should be considered only after working out a sound HFE plan. This way, adjustments needed to make the plan fit budget constraints can be made in light of the tradeoffs between costs and HFE technical benefits. The HFE schedule must be based on the system development schedule so that HFE work can be coordinated with system development activities in a manner that makes it possible to influence system design without delaying system design activities.

These four sets of information provide the database needed for constructing the HFE plan. Construction of the plan is described in the next section.

5.1.3 Constructing the HFE Plan

Constructing the HFE plan consists of three tasks:

1 developing the HFE plan's technical content,

2 defining how the HFE plan will be integrated with the overall system development plan, and

3 defining the HFE plan's project management mechanisms.

The following subsections describe these three tasks.

5.1.3.1 HFE Plan Technical Content

The CCIS HFE Requirements DID titled Human Factors Engineering Plan provides a ready-made detailed outline for the technical content of an HFE plan. Thus, the HFE plan for a specific project can be produced by filling in this outline with descriptions of how the various HFE tasks will be performed.

Two additional DIDs may be used to make up the HFE plan. These are the Human Factors Engineering Test Plan and the Human Factors Engineering Simulation Plan. These DIDs are intended for describing detailed plans for running HFE tests, and for doing any substantial simulation studies. Normally the data items described in these DIDs will be submitted after the project starts when enough information available to do detailed planning for design tests and simulation work. However, these DIDs can be used to help prepare the sections of the Human Factors Engineering Plan for summarizing intentions for conducting HFE tests and simulation studies. The Human Factors Engineering Test Plan and the Human Factors Engineering Simulation Plan can then be prepared and submitted at the appropriate times during the project.

Although the three DIDs just discussed are the only ones addressed to planning, reviewing the contents of all the remaining DIDs will nevertheless be important. The remaining DIDs provide detailed outlines for reporting the results of all the HFE work. As such, they describe what has to be accomplished in each HFE task. They are therefore good reminders of the tasks that must be performed during the execution of the HFE plan.

The DIDs provide a good foundation for planning the technical content of the HFE plan, but they do not suggest how to integrate the HFE plan with the overall system development plan. The next section describes how to do this.

5.1.3.2 Integrating the HFE Plan with the System Development Plan

As explained earlier in Section 3.2.1.4.2, (page 19), when the PM designs a project, he or she usually specifies a particular system engineering method to be used in the system development project. Since system engineering methods do not normally contain significant HFE components or mechanisms for HFE inputs, the HFE plan must always include provisions for integrating HFE

activities with the overall system development plan. The discussion in CCIS HFE Requirements User Manual (5.3.2.2.1) should be helpful for deciding how to integrate the HFE requirements with the system development approach.

The CCIS HFE Requirements User Manual (5.3.2.2.1) discusses the main types of system development approaches currently in use, including special cases such as iterative development approaches, small applications development, and adapting off-the-shelf systems. The discussion concludes that a CCIS HFE Requirements HFE plan can be integrated into almost any system development approach if three requirements are met:

1 the HFE analysis and design testing activities are scheduled so that their results can influence system design;

2 the detailed design of system components that support user tasks reflects the application of HFE design requirements, and

3 the HFE analysis of users' tasks in the system design is kept current with the design.

Applying these principles is generally practical because system engineering approaches seldom include any significant HFE content. Therefore, the project's system engineering method will not likely contain anything that would create technical conflicts with the HFE methods represented in the CCIS HFE Requirements. In cases where a system engineering method contains HFE-like activities, it should be possible to replace or augment them with CCIS HFE Requirements equivalent activities.

To define how the HFE plan is going to integrate with the system development plan in a particular project, the HFE planner will need to be, or become familiar with two things: 1) the system engineering method that will be used in the project, and 2) the overall system engineering plan as it emerges during the project. The planner must be familiar with the system engineering method to be able to understand the technical relationships between HFE activities and system engineering activities. The planner must be familiar with the system development plan to be able to synchronize HFE activities with other system engineering activities. If the planner understands the project's system engineering approach and the system development plan, integrating HFE activities with other system development activities should present little difficulty.

5.1.3.3 Integrating HFE with the Project Management Plan

The project management method is the system that the PM uses to manage execution of a CCIS acquisition project. As explained earlier in Section 3.2 (page 12) a system development project management system is essentially a set of specifications for how the contractor will report plans for work and report progress on work. The PM will specify a project management standard for the project as part of the project requirements. However, like the available system engineering methods, project management standards rarely contain provisions for HFE. This is why the CCIS HFE Requirements was created, and why it is essentially a project management tool for managing HFE activities. Thus, the PM can specify the HFE project management requirements by specifying which CCIS HFE Requirements DIDs are to be used for reporting HFE plans and work

progress. The CCIS HFE Requirements User Manual (5.3.2.2.2) advises the PM on how to specify the DIDs for a particular project.

The CCIS HFE Requirements User Manual (5.3.2.2.2) describes the contents of the CCIS HFE Requirements DIDs, and how to specify the DIDs for a specific project. This section also provides several model statements for specifying the use of the DIDs as project management media. For example, the following statement specifies HFE deliverables and project management requirements.

> `The conduct and management of the HFE program shall conform to applicable sections of the CCIS HFE Requirements, and an HFE plan prepared and maintained in accordance with the CCIS HFE Requirements DID titled HFE Plan. The HFE program deliverables shall include the following data items as specified in the CCIS HFE Requirements DIDs: HFE Plan, HFE Test Plan, HFE Simulation Plan, HFE Analysis Report, HFE System Requirements and Function Allocation Report, HFE Design Approach Document, HFE Progress Report, HFE Test Report, HFE Design Review Report'

When the CCIS HFE Requirements is specified for a project, a statement to this effect should appear in the RFP for the project.

Another statement that may appear in the RFP is the following from CCIS HFE Requirements User Manual (5.3.2.2.2), which specifies use of the Human Factors Engineering Plan DID for describing HFE plans in proposals.

> `The bidder's proposed system engineering plan shall contain an HFE Plan conforming to the CCIS HFE Requirements DID titled HFE Plan. The proposed HFE Plan will form the project's baseline HFE Plan. The HFE Plan shall include a description of the work covered by each of the CCIS HFE Requirements Document DIDs specified in the project statement of work. If any data items specified by the CCIS HFE Requirements Document DIDs are to be produced in more than one version, for example, as extensions or up-dates of prior versions, the separate versions shall be identified in the HFE Plan. The HFE Plan schedule shall identify the time when each deliverable is to be submitted. The HFE Plan shall also show how the HFE Plan will be synchronized with other system engineering activities.'

Tables 18 and 19 in the CCIS HFE Requirements User Manual give additional model statements for specifying HFE project management requirements for different types of development projects. Although HFE project management requirements can be stated in various ways, they all specify the use of some combination of the CCIS HFE Requirements DIDs.

If the project is to include system implementation assistance, the SOW should contain a statement specifying the HFE reporting requirements for implementation. The CCIS HFE Requirements User Manual (5.3.2.4) describes how to specify these requirements. The section suggests using the Human Factors Engineering Progress Report DID for reporting on HFE activities in implementation.

For the purposes of the HFE planning, once the DIDs have been specified in the RFP, the HFE planner only needs to confirm that the specified DIDs will be used for reporting, and then schedule the times when the reports and deliverables are to be completed. The HFE reports and DIDs can then be fitted into the overall project management reporting system either by cross-references or as supplementary reports.

5.1.3.4 Constructing the HFE Budget and Schedule

The next two subsections treat budgeting and scheduling separately.

5.1.3.4.1 Budgeting for the HFE Plan

Everyone has their own approach to working out a budget for a project, and this document does not presume to suggest how to construct a budget. However, no matter how the budget is approached, the following three areas deserve special attention.

1 *Task analysis budget.* The time required to do a task analysis of users' tasks in an existing system can be highly variable. The accuracy of the estimated time to do a task analysis largely depends on how much prior information one has about the jobs to be analysed. If the task analysis work represents a substantial proportion of the work, it is worth trying to get as much prior information as possible about the jobs to be analysed, and what will be involved in doing the analysis. This might include, for example, reading up on the jobs, asking to see the operations, consulting with retired personnel who know the operations, and so on.

2 *DID reports.* The reporting requirements in the CCIS HFE Requirements DIDs are considerably more extensive than many HFE specialists are used to. To avoid underestimating the time required to write the DID reports, all the applicable DIDs should be reviewed with a focus on how much time will be required to write up each report. It also needs to be remembered that most of the DIDs can represent several reports corresponding to different phases of the project.

3 *Costs of HFE Plan support items.* Although by far the greatest proportion of the HFE budget will be for direct HFE specialist labour, there may still be substantial expense items that have to be in someone's budget, if not the HFE budget. For example, the costs of system representations such as mockups and prototypes have to be accounted for somewhere in the project budget. Whether these costs go into the HFE budget or another budget does not matter. The important thing is to ensure that these costs are estimated and accounted for, and that it is not just assumed that another system engineering budget item has funds to cover them. To avoid later finding that there are no funds available for equipment or services needed to support the HFE plan, the plan should be reviewed to identify such support items. This will ensure that support costs are covered in the project budget.

The three areas just described are ones that will usually deserve special attention. Since they may not be the only ones, the HFE plan should be reviewed to see if the plan has any other areas where it might be particularly vulnerable to budget problems.

A final potential issue in HFE budgeting is justification for the cost of the HFE effort. This issue is typically raised as a question about the cost-benefit of HFE. If the question cannot be avoided, Rouse and Boff [116] provides some ammunition for dealing with the question. Generally the benefits of HFE are the ones well known to HFE specialists, that is,

1 solving design problems that could not be solved without HFE,

2 achieving human performance not otherwise achievable,

3 improved human and system performance,

4 enhanced customer and user acceptance,

5 avoidance of costs associated with inadequate human performance or performance errors, and

6 increased confidence in the system.

These benefits may all be tangible to varying degrees, but they are usually difficult to quantify. So, while one can list the benefits, inability to quantify the benefits leaves the cost-benefit question moot.

Perhaps the best way deal with the cost-benefit question is to deal with the logical fallacy that underlies it. Typically the question of the cost-benefit of HFE begs the question: what is the cost-benefit of any system engineering effort? Indeed, what is the cost-benefit of a CCIS, or the cost-benefit of having military forces? Questions about the cost-benefit of public expenditures usually turn out to be imponderables. This is because the benefits are generally matters of social values that do not translate into currency values without making assumptions that many people are not willing to make (Chapter. 14, [111]).

A CCIS is a key system for supporting the humans who comprise the command and control organization. Whether or not to spend money on HFE in CCIS development is ultimately a judgment about the value of ensuring that the CCIS will have the verifiably sound HFE qualities needed to support the people in the command and control organization.

5.1.3.4.2 HFE Plan Scheduling

To repeat what was said a little earlier, to influence system design, various HFE activities must be completed at the times when their results will be needed as system design inputs. Nearly all HFE activities can be done in parallel with other system engineering activities. The activities also take about the same times to complete as corresponding system engineering activities. Therefore, synchronizing the HFE plan with the rest of the system development plan should be easy to do. In particular, scheduling the activities in the HFE Plan can be done by taking the overall system development schedule and timing the HFE activities according to events in the system development schedule.

5.1.3.5 Summary of Constructing the HFE Plan

The following points summarize construction of the HFE Plan.

1 The Human Engineering Plan DID provides a detailed outline for developing the technical content of the HFE Plan. Where needed, the Human Factors Engineering Test Plan and the Human Factors Engineering Simulation Plan DIDs can be used to describe detailed plans for HFE tests and simulation studies.

2 Besides its technical content, the HFE plan must include provisions for integrating HFE work with the overall system development plan. Generally the HFE Plan can be integrated with any system development approach if

   2.1 the HFE analysis and design testing activities are scheduled so that their results can influence system design;

   2.2 the detailed design of system components that support user tasks reflects the application of HFE design requirements, and

   2.3 the HFE analysis of users' tasks in the system design is kept current with design.

3 The HFE plan must also include provisions for integrating the management of the HFE Plan with the overall project management approach specified for the project. The CCIS HFE Requirements DIDs are the primary vehicles for defining HFE reporting and deliverables. The project RFP will normally specify the DIDs to be used for HFE project management.

4 Budgeting for the HFE Plan activities does not present any extraordinary problems, but three areas may need extra attention: 1) accurate estimation of effort needed for task analysis work, 2) ensuring that adequate time is allocated for completing DID reports, and 3) ensuring that costs for equipment and services needed to support HFE activities are accounted for in the project budget.

5 Scheduling of the HFE work will be determined by the system development plan schedule.

5.2 Managing Execution of the HFE Plan

The HFE specialist has three responsibilities in the management of the HFE Plan:

1 to ensure that HFE work is done properly, within budget, and on time;

2 to complete and submit the project's DID reports and deliverables on time, and

3 to cooperate with the PM, PMO staff, and contractor project managers in project management tasks associated with the HFE work.

The first two responsibilities are the ordinary responsibilities of doing the HFE work. The third responsibility generally means cooperating with the PM in the PM's project management tasks connected with managing the execution of the HFE Plan.

The CCIS HFE Requirements User Manual (5.4) describes the PM's project management tasks. From the summary this User Manual section, the PM's tasks are the following.

1 Launch the HFE Plan by ensuring that the plan satisfies project requirements.

2 Maintain control over the HFE budget.

3 Maintain the HFE Plan schedule.

4 Manage HFE tests and system acceptance trials.

5 Manage system implementation.

6 Monitor HFE reports and deliverables.

7 Conduct project progress reviews with users and stakeholders.

If the HFE Plan is well designed and competent people will do the work, none of these project management tasks impose any particularly onerous duties on HFE personnel. The following sections describe the roles of HFE personnel in each of these tasks.

5.2.1 Launching the HFE Plan

Before the start of the project, the PM will review each system engineering plan to ensure that it will satisfy the project requirements. This is particularly important when the contractor was selected in a competitive bidding process. In a competitive bidding environment, the selected contractor presumably has the best overall proposal, but not necessarily the best proposal in every aspect of the project. In particular, the contractor may have been selected in spite of perceived weaknesses in the HFE Plan. If so, the PM will very likely ask the contractor to modify the HFE Plan. If significant changes are needed, the PM will specify the changes. The CCIS HFE Requirements User Manual (5.4.1) describes the PM's options for modifying the HFE Plan.

Even if the HFE Plan is generally sound, modifications may still be needed so the plan can respond to events that have occurred since the project RFP was issued; respond, for example, to changes in the user organization's requirements. Thus, at the beginning of the project, the HFE planner must be prepared to make some modifications to the HFE Plan. These modifications can range from being minor modifications to major modifications requiring a good deal of work. Obviously, the simplest way to avoid having to make major plan modifications is to put in a solid effort at constructing the plan in the first place.

5.2.2 Controlling the HFE Budget

During the project, the PM monitors the HFE budget to ensure that expenditures stay within the plan, and remain consistent with the progress of the work. The CCIS HFE Requirements User Manual (5.4.2) describes the main factors that the PM needs to take account of in HFE budget control. These are

1 the precision of the system development requirements;

2 the accuracy of the HFE Plan estimates of levels of HFE effort required for different activities;

3 the inclusion of iterative development methods in the system development;

4 the timeliness and accuracy of reporting on HFE activities, and

5 reallocating HFE budget to other system engineering activities.

The need for precision in defining requirements is emphasized throughout both the CCIS HFE Requirements User Manual and this CCIS HFE Design Guidelines. If the need has been taken seriously during HFE planning, it should not become a problem in budget management.

As the CCIS HFE Requirements User Manual (5.4.2) says, modest deviations from the planned HFE budget expenditures are to be expected, and should not be a problem if over expenditures are balanced by under expenditures. When there are impending overruns on budget items, the responsible HFE specialists must keep project managers informed about the reasons for the deviations, and be ready to recommend ways to compensate for the overruns. Modest variations from budget plans are normal in any project. They only become a problem when they are ignored and turn out to be symptoms of an undetected deeper budget problem.

Iterative development methods are notorious for causing budget overruns. The means to control, and if possible, to avoid iterative development methods were discussed earlier in Section 4.3.1.5 (page 141), which also contains cross-references to discussions of iterative development in the CCIS HFE Requirements User Manual.

The timeliness and accuracy of reporting on HFE activities are important to project management budget control because reports on activities are the PM's primary means of judging whether or not the progress of work is consistent with budget expenditures. If reports of progress are inaccurate or behind time, problems may go undetected past the time when action can be taken to correct the problems.

Reallocation of the HFE budget can become necessary when other system engineering activities run into unforseen difficulties. When this happens, non-HFE authorities will be tempted to cut back on analysis activities and retain HFE testing activities because few acquisition authorities are willing to accept a system without a system design test. Consistent with the position evident throughout all the CCIS HFE Requirements documentation, trying to conduct design tests without supporting HFE analysis information is generally a waste of time. Thus, the CCIS HFE Requirements User Manual (5.4.2.4) advises the PM against reallocating the HFE budget by just cutting back analysis activities. Instead, cutting back all the HFE activities in proportion is usually better than cutting back a particular activity. Overall, a balanced, if limited, HFE plan will be more effective than an HFE plan devoted to just one activity such as design testing.

5.2.3 Controlling the HFE Schedule

Generally, controlling the HFE schedule should not present any problems provided the HFE work is kept synchronized with the overall system development schedule. However, there is one HFE activity to which the PM needs to pay particular attention. This is design testing that involves participants from the user organization. The CCIS HFE Requirements User Manual (5.4.3) advises the PM to review carefully the HFE reports to detect and respond to potential problems associated with getting design tests done on time. This implies that particular care must be taken to see that HFE reports on design testing plans give the PM a complete basis for spotting any impending scheduling problems connected with design tests.

5.2.4 Managing Design Tests and the System Acceptance Trial

The PM is responsible for coordinating all user organization participation in HFE testing activities. For their part, HFE personnel can best help the PM in this task by:

1 making sure that the HFE reports are timely and complete,

2 providing information that will let the PM arrange user organization participation well ahead of time, and

3 cooperating in the day-to-day liaison activities that are part of coordinating a design test.

Data analysis is one component of design testing that can sometimes hold up other work. Therefore, it is a good idea to keep project management well informed about data analysis progress, and to be prepared to provide interim results, or make other accommodations that would avoid delays in other work.

5.2.5 Manage System Implementation

Presuming that HFE implementation activities are covered in the HFE Plan, this project management responsibility does not require any special attention not required by any other HFE activity. The section in the CCIS HFE Requirements User Manual concerned with managing implementation points out the need for coordination with the user organization's implementation plan. It also points out that more user organization personnel will be involved in implementation than there were during system development.

5.2.6 Monitor HFE Reports and Deliverables.

This is a project management task that requires no HFE action beyond preparing the reports and deliverables.

5.2.7 Conduct Progress Reviews

In all projects, the PM maintains a review process that keeps all the parties to the acquisition informed about the project's progress. The CCIS HFE Requirements User Manual (5.4.9) describes this task. The DID reports should give the PM all the information on HFE activities needed for conducting progress reviews. Section 5.4.9 points out that DIDs reporting on design reviews and design tests are an important source of information for judging the effectiveness of the emerging CCIS design. Because of this use, 5.4.9 also points out that HFE design test reports should be written for a wider readership, and provide a concise picture of how well the system is progressing towards meeting the design requirements.

Progress Reviews normally include progress review meetings. These may require preparation of presentations of the results of HFE activities. Usually requirements for participation in review meetings will be specified in the project RFP, and can therefore be planned for in the HFE plan or the contractor's project management plan.

5.2.8 Summary of HFE Project Management Responsibilities

The following summarizes HFE responsibilities in project management.

1 Cooperate in launching the HFE plan. This may include adjusting or modifying the plan to meet the project requirements fully or to take account of circumstances that have changed since the plan was created.

2 Provide complete and timely reports that will keep the PM fully informed about the progress of HFE work relative to budget expenditures, and when necessary, cooperate in modifying HFE plans to respond to budget shortfalls.

3 Keep the PM informed about any circumstances that could potentially affect the HFE schedule, and provide information that the PM needs for ensuring that arrangements for user organization participation can be made well in advance.

4 Cooperate with the PM in coordinating participation in design test and acceptance trials.

5 Cooperate as needed in the management of implementation activities.

6 Ensure that HFE reports and deliverables support the PM's responsibilities for monitoring the progress of the project.

7 Ensure that HFE reports, particularly design review and design test reports provide timely and accurate information for design reviews. When required, participate in design review meetings.

6 REFERENCES

1. Engel, R. & Townsend, M. *Guidelines for Human Factors Engineering Requirements for Canadian Forces Command and Control Information Systems.* For Defence and Civil Institute of Environmental Medicine by Engel and Townsend, Toronto, Ont. April, 1996.

2. U.S. Department of Defense. *Human Engineering Requirements for Military Systems, Equipment, and Facilities (MIL-STD-46855).* U.S. Department of Defense, Washington, DC, 1994.

3. Engel, R. & Townsend, M. *User Manual: Guidelines for Human Factors Engineering Requirements for Canadian Forces Command and Control Information Systems.* For Defence and Civil Institute of Environmental Medicine by Engel and Townsend, Toronto, Ont. April, 1998.

4. Norman, Donald A. *The Psychology of Everyday Things.* Basic Books, Inc., New York, 257p, 1988.

5. Meister, David. *Conceptual Aspects of Human Factors.* The Johns Hopkins University Press, Baltimore, MD, 263, 1989.

6. Chapanis, Alphonse. *Human Factors in Systems Engineering.* John Wiley & Sons, Inc. New York. pp. 332, 1996.

7. Salvendy, Gavriel, Ed. *Handbook of Human Factors and Ergonomics.* John Wiley & Sons, 2137p, 1997.

8. Salvendy, Gavriel, Ed. *Handbook of Human Factors.* John Wiley & Sons, 1874p, 1987.

9. Kantowitz, Barry. H., & Sorkin Robert D. *Human Factors: Understanding People-System Relationships.* John Wiley & Sons New York, 700p, 1983.

10. Brown, C. Marlin "Lin". *Human-Computer Interface Design.* Ablex Publishing Corp. Norwood, NJ, 236p, 1988.

11.Fuller, J. F. C. *A Military History of the Western World,* Vols. I to III. Da Capo Press, New York, 1987 (original publication 1954).

12. U.S. Department of Defense. *Human engineering design criteria for military systems, equipment, and facilities (MIL-STD 1472D).* U.S. Department of Defense, Washington, DC, March, 1989.

13. Open Software Foundation. *OSF/Motif Style Guide, Revision 1.0.* Open Software Foundation, Cambridge, Massachusetts, 1989.

14. Avery, Larry W. & Bowser, Samuel E. (Eds.). *Human Factors Design Guidelines for the Army Tactical Command and Control System (ATCCS) Soldier-Machine Interface: Version 2.0.* Prepared by Battelle Pacific Northwest Laboratory, U.S. Dept. of Energy, for U.S. Army Tactical Command and Control System Experimentation Site, Fort Lewis, Washington, May, 1992.

15. CTA Incorporated. *User-Interface Guidelines.* Data Systems Technology Division, Goddard Space Flight Center, January, 1996.

16. U. S. Department of Defense. *Department of Defense Handbook: Human Engineering Guidelines for Management Information Systems (MIL-HDBK-761A)*. U.S. Department of Defense, Washington, DC, 272p. September, 1989.

17. Engel, R. & Townsend M. *Guidelines for the Design and Evaluation of Operator Interfaces for Computer Based Control Systems*. For Defence and Civil Institute of Environmental Medicine under Contract # 09SE.W7711-5-9362/01-SE. Engel and Townsend, Toronto, Ont. 1989.

18. Beevis, D., Bost, R. Döring, B., Nordø, E., Oberman, F., Papin, J-P, Streets, D., & Schuffel, H. *Analysis Techniques for Man-Machine System Design, Vol. 1*. NATO Defence Research Group Technical Report Ac/243(Panel 8)TR/7, Defence and Civil Institute of Environmental Medicine, Downsview, Ontario, Canada, 104 p., July, 1992.

19. Engel, R. & Townsend, M. *Division Command and Control Performance Measurement.* For Defence and Civil Institute of Environmental Medicine under Contract # 09SE.W7711-7-7013. Engel and Townsend, Toronto, Ont. 1991.

20. Meister, David. *Systems design, development, and testing. In Handbook of Human Factors, G Salvendy (Ed.)*. John Wiley & Sons, New York, 1874p, 1987.

21. U.S. Department of Defence. *Military Standard Defense System development DOD-STD-2167A.* U.S. Department of Defence, Washington, 29 February 1988.

22. U.S. Department of Defence. *Software Development and Documentation DOD-STD-498.* U.S. Department of Defence, Washington, 1994.

23. Andersen Consulting. *Method/1: Guide for System Users*. Andersen Consulting, Arthur Andersen & Co., S.C.

24. Luczak, Holger. *Task Analysis: in Handbook of Human Factors and Ergonomics, G Salvendy (Ed.).* John Wiley & Sons, New York, 2137p, 1997.

25. Laughery, Kenneth R., & Laughery, K. Ronald. *Analytic Techniques for Functional Analysis: in Handbook of Human Factors, G Salvendy (Ed.).* John Wiley & Sons, New York, 1874p, 1987.

26. Buffa, Elwood S. *Operations Management: the Management of Productive Systems.* John Wiley & Sons, Santa Barbara, USA, 686p 1976.

27. Holley, I. B. Jr. *Command, Control, and Technology.* Defense Analysis, Vol 4, No 3, pp 267-286, 1988.

28. Sharit, Joseph. *Allocation of Functions: in Handbook of Human Factors and Ergonomics, G Salvendy (Ed.).* John Wiley & Sons, New York, 2137p, 1997.

29. Ross, D.T. & Schoman, K.E. Jr. *Structured analysis for requirements definition.* IEEE Transactions in Software Engineering, SE-3 (1), pp. 6-15, January, 1977.

30. Marca, D.A., & McGrowan, C.L. *SADT™ - Structured analysis and design technique.* New York: McGraw-Hill Book Co., New York, 1987.

31. Ross, D.T. *Applications and extensions of SADT.* IEEE Computer Magazine, Vol. 18, No. 4, pp 25-34, 1987.

32. Luczak, Holger. *Task Analysis: in Handbook of Human Factors and Ergonomics, G Salvendy (Ed.).* John Wiley & Sons, New York, 2137p, 1997.

33. Engel, R. & Townsend, M. *Division Command and Control Performance Measurement.* For Defence and Civil Institute of Environmental Medicine under Contract # 09SE.W7711-7-7013. Engel and Townsend, Toronto, Ont. 1991.

34. Boettcher, K. L. & Levis, A. H. *Modelling the Interacting Decision-Maker with Bounded Rationality.* IEEE Transactions on Man, Systems, and Cybernetics, Vol SMC-12, No. 3, pp 334-344, 1982.

35. Athans, Michael, Ducot, Elizabeth R., Levis, Alexander H. & Tenney, Robert R. *Proceedings of the 6th MIT/ONR Workshop on C³ Systems.* Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, Mass. Report # LIDS-R-1354, 1983.

36. Grevet, Jean-louis M. *Decision Aiding and Coordination in Decision-Making Organizations.* Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, Mass. Report # LIDS-TH-1737, 1988.

37. Cothier, Philippe H. & Levis, Alexander H. *Timeliness and Measures of Effectiveness in Command and Control.* IEEE Transactions on Man, Systems, and Cybernetics, Vol SMC-16, No. 6, pp 844-853, 1986.

38. Levis, Alexander H., Tomovic, M. M. & Cothier, P. M. *Information Processing Organizations with Acyclical Information Structures.* U. S. Army Research Institute for the Behavioral and Social Sciences, Alexandria, VA, ARI Research Note 87-42, 1987.

39. Chyen, Gloria H.-L. & Levis, Alexander H. *Analysis of Preprocessors and Decision Aids in Organizations.* Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, Mass. Report # LIDS-P-1438, 1985.

40. Levis, Alexander H. *Modelling and Measuring Effectiveness of Cy³ Systems.* Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, Mass. Report # LIDS-P-1608, 1986.

41. Engel, R. & Townsend, M. *Increasing Marine Traffic Regulator Capacity.* Engel & Townsend, Toronto, Ontario: for Transportation Development Centre, Montreal, Quebec, Report No. TP11681E, June 1993.

42. Andersen Consulting. *Method/1: Guide for System Users.* Andersen Consulting, Arthur Andersen & Co., S.C.

43. Miller, R.B. *Task taxonomy: Science or technology.* Ergonomics, Vol. 10, pp167-176, 1967.

44. Drury, Colin G., Paramore, Barbara, Van Cott, Harold P., Grey, Susan M., & Corlett, E. Nigel. *Task Analysis: in Handbook of Human Factors, G Salvendy (Ed.)*. John Wiley & Sons, New York, 1874p, 1987.

45. Annett, J. & Duncan, K. D. *Task analysis and training design*. Occupational Psychology, Vol 41, pp211-221, 1967.

46. Nielsen, Jakob. *Usability Testing: in Handbook of Human Factors and Ergonomics, G Salvendy (Ed.)*. John Wiley & Sons, New York, 2137p, 1997.

47. Chubb, Gerald P., Laughery, Jr., K. Ronald, & Pritsker, A, Alan B. *Simulating Manned Systems: in Handbook of Human Factors, G Salvendy (Ed.)*. John Wiley & Sons, New York, 1874p, 1987.

48. Laughery, Jr., K. Ronald & Corker, Kevin. *Computer Modeling and Simulation: in Handbook of Human Factors and Ergonomics, G Salvendy (Ed.)*. John Wiley & Sons, New York, 2137p, 1997.

49. McMillan, G. R., Beevis, D., Salas, E., Strub, M. H., Sutton, R. & van Breda, L. *Applications of Human Performance Models to System Design*. Plenum Press, New York, Defense Research Series, Volume 2, 1989.

50. Wickens, C. D. *The structure of attentional resources. In R. S. Nickerson, Ed., Attention and Performance VIII*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1980.

51. Tsang, Pamela, & Wilson, Glenn F. *Mental Workload: in Handbook of Human Factors and Ergonomics, G Salvendy (Ed.)*. John Wiley & Sons, New York, 2137p, 1997.

52. Wickens, Christopher, D. & Carswell, C. Melody. *Information Processing: in Handbook of Human Factors and Ergonomics, G Salvendy (Ed.)*. John Wiley & Sons, New York, 2137p, 1997.

53. Buffa, Elwood S. *Operations Management: the Management of Productive Systems*. John Wiley & Sons, Santa Barbara, USA, 686p 1976.

54. Cameron, F. W. P. *Command and Control Effectiveness and Critical Path Analysis*. DLOR Staff Note 90/4, Directorate of Land Operational Research, Department of National Defence, Ottawa, Canada. 1990.

55. Engel, G. R. & Townsend, M. *Evaluation of the Tactical Army Command and Control Data System (TACCDAS) Test Series 2 Evaluation Trials*. Defence and Civil Institute of Environmental Medicine, Toronto, Ontario, Jan. 1986.

56. Sage, Andrew P. *Decision support systems: in Handbook of Human Factors and Ergonomics, G Salvendy (Ed.)*. John Wiley & Sons, New York, 2137p, 1997.

57. Kantowitz, Barry H. & Sorkin, Robert D. *Allocation of functions. In Handbook of Human Factors, G Salvendy (Ed.)*. John Wiley & Sons, New York, 1874p, 1987.

58. Yourdon, E. *Modern structured analysis*. Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1989.

59. Beevis, D. *Experience in the Integration of Human Engineering Effort with Avionics Systems Development.* NATO Advisory Group for Aerospace Research and Development, Conference Proceedings No. 417, 1987.

60. McCormick, Ernest J. *Human Factors Engineering.* McGraw-Hill, New York, 639p., 1970.

61. Jordan, N. *Allocation of functions between man and machine in automated systems.* Journal of Applied Psychology, Vol. 47, pp161-165, 1963.

62. Meister, David. *Behavioral Analysis and Measurement Methods.* John Wiley, New York, 1985.

63. U.S. Department of Defence. *Human Engineering Procedures Guide (DoD-HDBK-763).* U.S. Department of Defence, Washington, 1987.

64. Lehto, Mark R. *Decision making: in Handbook of Human Factors and Ergonomics, G Salvendy (Ed.).* John Wiley & Sons, New York, 2137p, 1997.

65. Woodson, W. E. *Human Factors Design Handbook.* McGraw-Hill Book Co., 1981.

66. Beevis, D. *Bridge design – a human engineering approach in Canada.* Proceedings, Sixth Ship Control System Symposium, Vol 2. National Defence Headquarters, Ottawa, Canada, 1981.

67. Nielsen, J. *Usability Engineering.* Academic Press, New York, 1993.

68. US Department of Defense. *Human Computer Interface Style Guide, Version 2.* Defense Information Systems Agency, September 1992.

69. TRW Inc. *The AWIS Common User/System Interface Style Guide.* Prepared for U.S. Army WWMCCS Information System Project Management Office TRW Inc., Fairfax, VA, February, 1990.

70. U. S. Department of Defense. *User/Computer Interface (MIL-STD 1801 (USAF)).* U.S. Department of Defense, Washington, DC, May, 1987.

71. DMR Group Inc. *Land Force Command and Control Information System User Interface Standards.* DMR Group Inc. for Department of National Defence (DND File: 1150-110/L23 (DLR 4-8)), March, 1993.

72. de Souza, F., & Bevan N.. *The use of guidelines in menu interface design: Evaluation of a draft standard.* In Diaper, D., Gilmore D., Cockton, G., Shackel, B. Eds. Human-Computer Interaction - Interact'90, Participants Edition, pp 435-440, North-Holland, 1990.

73. Tetzlaff, L. & Schwartz, D.. *The use of guidelines in interface design.*. CHI '92 Conference Proceedings, Association of Computing Machinery, May, 1992.

74. Liu, Yili. *Software-user Interface Design: in Handbook of Human Factors and Ergonomics, G Salvendy (Ed.).* John Wiley & Sons, New York, 2137p, 1997.

75. Bullinger, Hans-Jorg, Kern, Peter, & Braun, Martin. *Controls: in Handbook of Human Factors and Ergonomics, G Salvendy (Ed.).* John Wiley & Sons, New York, 2137p, 1997.

76. Campbell, Gweneth U., & Druce, Guy L. *Assessment of desktop icons for Land Forces CCIS: Final Report.* Unger Campbell & Associates, Delta, BC. Report under contract # W8477-5-CB13/01-SV, 1996.

77. McMillan, Grant R., Eggleston, Robert G. & Anderson, Timothy R. *Nonconventional Controls: in Handbook of Human Factors and Ergonomics, G Salvendy (Ed.).* John Wiley & Sons, New York, 2137p, 1997.

78. Helander, Martin G. *Design of Visual Displays: in Handbook of Human Factors, G Salvendy (Ed.).* John Wiley & Sons, New York, 1874p, 1987.

79. Grandjean, Etienne. *Design of VDT Workstations: in Handbook of Human Factors, G Salvendy (Ed.).* John Wiley & Sons, New York, 1874p, 1987.

80. Smith, Michael J., & Cohen, William J. *Design of Computer Terminal Workstations: in Handbook of Human Factors and Ergonomics, G Salvendy (Ed.).* John Wiley & Sons, New York, 2137p, 1997.

81. Evans, S., Walker, C., & Beevis, D. *A Human Engineering Evaluation of DDH280 Bridge Design Proposals.* DCIEM No. 84-R-43. Toronto: Defence and Civil Institute of Environmental Medicine, 1984.

82. Hendy, K. *A model for human-machine-human interactions in workspace layout problems.* Human Factors Vol. 31, No. 5, 1989.

83. Ramussen, J., Pejtersen, A-M., & Goodstein, L. *Cognitive Engineering, Concepts and Applications.* John Wiley & Sons Inc., New York. 1995.

84. Moray, Neville. *Human Factors in Process Control: in Handbook of Human Factors and Ergonomics, G Salvendy (Ed.).* John Wiley & Sons, New York, 2137p, 1997.

85. Bond. Jr., Nicholas A. . *Maintainability. In Handbook of Human Factors, G Salvendy (Ed.).* John Wiley & Sons, New York, 1874p, 1987.

86. Tilley, K. W. *Fault diagnosis for maintenance personnel.* Ergonomics 10 (1) 206-213, 1967.

87. Garg-Janardan, C., Eberts, R. E., Zimolong, B., Nof, S. Y., & Salvendy, S. *Expert Systems: in Handbook of Human Factors, G Salvendy (Ed.).* John Wiley & Sons, New York, 1874p, 1987.

88. Geiselman, R. Edward, Landee-Thompson, Betty, & Samet, Michael G. *A Sellective Callup System for Managing Tactical Information on Graphic Displays.* IEEE Transactions on Systems, Man, and Cybernetics, Vol SMC-16, No 6, pp 901-908, 1986.

89. Defense Systems Inc. *Headquarters Effectiveness Program Summary.* U.S. Defense Communications Agency, Washington, 1983.

90. Defense Systems Inc. *Correlation of Experimental and Exercise Results.* U.S. Defense Communications Agency, Washington, DC, #DCA100-86-C-0004, 49p, 1986.

91. Wiener, Earl L. *Cockpit Automation.* E. L. Wiener & D. C. Nagel, Eds. Human Factors in Aviation. Academic Press Inc., 684p, 1988.

92. Sheridan, Thomas B. *Supervisory Control: in Handbook of Human Factors and Ergonomics, G Salvendy (Ed.).* John Wiley & Sons, New York, 2137p, 1997.

93. Foster, Gregory D. *Contemporary C^2 Theory and Research: the Failed Quest for a Philosophy of Command.* Defense Analysis, Vol. 4, No 3, pp 201 228, 1988.

94. Command and Control Division Winter Study Group, USAF Air Research and Development Command. *The challenge of command and control.* Air University Library M35060-1-C No. 4, 1961.

95. Andriole, Stephen J. *Leveraging Command and Control via Enhanced Command Decisionmaking: Prospects for a Behavioral Theory of Command and Control.* Defense Analysis, Vol 4, No 3, pp 253 265, 1988.

96. Sarter, N. B., Woods, D. D., & Billings, C. E. *Automation Surprises: in Handbook of Human Factors and Ergonomics, G Salvendy (Ed.).* John Wiley & Sons, New York, 2137p, 1997.

97. Malone, T. B. & Micocci, A. J. *Development and Validation of Methods for Man-Machine Interface Evaluation: Final Report.* National Aeronautics and Space Administration, NASA-CR-142829, Washington, DC, 1975.

98. Meister, David. *System Effectiveness Testing: in Handbook of Human Factors, G Salvendy (Ed.).* John Wiley & Sons, New York, 1874p, 1987.

99. Majoros, Anthony E., & Boyle, Edward. *Maintainability: in Handbook of Human Factors and Ergonomics, G Salvendy (Ed.).* John Wiley & Sons, New York, 2137p, 1997.

100. American National Standards Institute. *Guide to Human Performance Measurements (ANSI/AIAA G-035-1992).* American National Standards Institute, sponsored by the American Institute of Aeronautics and Astronautics, 1993.

101. Wickens, Christopher D. *Information Processing, Decision-Making, and Cognition: in Handbook of Human Factors, G Salvendy (Ed.).* John Wiley & Sons, New York, 1874p, 1987.

102. Cronbach, Lee J. *Essentials of Psychological Testing.* Harper & Row, New York, 1970.

103. American Psychological Association. *Standards for Educational and Psychological Tests and Manuals.* American Psychological Association, Washington, D.C., 40p., 1966.

104. Cronbach, Lee J. *Essentials of Psychological Testing.* Harper & Row, New York, 1990.

105. Salvendy, Gavriel, & Carayon, Pascale. *Data Collection and Evaluation of Outcome Measures: in Handbook of Human Factors and Ergonomics, G Salvendy (Ed.).* John Wiley & Sons, New York, 2137p, 1997.

106. Sarter, N. B., & Woods, D. D. *Pilot interaction with cockpit animation II: An experimental study of pilots' model and awareness of the flight management and guidance system.* International Journal of Aviation Psychology, Vol. 4, No. 1, pp1-28, 1994.

107. Gaines, Dr. B.R., & Moray, Dr. N. *Development of Performance Measures for Computer Based Man-Machine Interfaces.* Centre for Person Computer Studies Inc., Man-Machine Systems Consultants Inc. DCIEM-PER-FIN: July 1985.

108. Sudman, Seymour, & Bradburn, Norman M. *Asking Questions.* Jossey-Bass Publishers, San Francico, 397 pgs. 1983.

109. Lord, Frederic M., & Novick, Melvin R. *Statistical Theories of Mental Test Scores.* Addison-Wesley, Reading, MA, 1968.

110. Barton, A. J. *Asking the embarrassing question.* Public Opinion Quarterly, Vol. 22, 1958.

111. von Winterfeldt, Detlof, & Edwards, Ward. *Decision Analysis and Behavioral Research.* Cambridge University Press, Cambridge, UK, 1986.

112. Sudman, Seymour. *Applied Sampling.* Academic Press, New York, 249 pgs. 1976.

113. Schlaifer, Robert. *Probability and Statistics for Business Decisions.* McGraw-Hill, New York, 1959.

114. Kish, Leslie. *Survey Sampling.* John Wiley & Sons, New York, 1965.

115. Ferguson, George A. *Statistical Analyis in Psychology and Education.* McGraw-Hill Book Company, New York, 1966.

116. Rouse, William B. & Boff, Kenneth R. *Assessing Cost/Benefits of Human Factors: in Handbook of Human Factors and Ergonomics, G Salvendy (Ed.).* John Wiley & Sons, New York, 2137p, 1997.

#512489